
OneBusAway Multi-Region – Rapidly Expanding Mobile Transit Apps to New Cities

Sean J. Barbeau, Ph.D.

University of South Florida

Alan Borning, Ph.D.

University of Washington

Kari Watkins, Ph.D.

Georgia Institute of Technology

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 applications (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 (Washington) 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 also is examined.

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 percent shorter than riders who do not have access to real-time information (Watkins et al. 2011). Additionally, real-time information users save almost two minutes in actual wait 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 (Washington), Salt Lake City (Utah), Columbus (Ohio), and Providence/Kingston (Rhode Island), concluded that real-time information attracts new riders who are otherwise reluctant to start using transit (Cluett et al. 2003). Similarly, a study in Chicago found modest ridership gains from real-time information even prior to wide usage of smart phones (Tang and Thakuriah 2012). Interviews with transit riders in San Francisco and Seattle in 2010 revealed that when the real-time information system was down, some riders elected not to ride the bus (Steinfeld and Zimmerman 2010). Riders also can 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 (Zimmerman et al. 2011). Other benefits identified in surveys include increased walking (i.e., public health benefits) and, for some riders, increased feelings of safety while waiting, particularly at night (Ferris et al. 2010; Gooze et al. 2013). With the number of smartphone users among transit riders being similar to those in the general population, providing app-based real-time information could be a major benefit to a large proportion of riders (Windmiller et al. 2014).

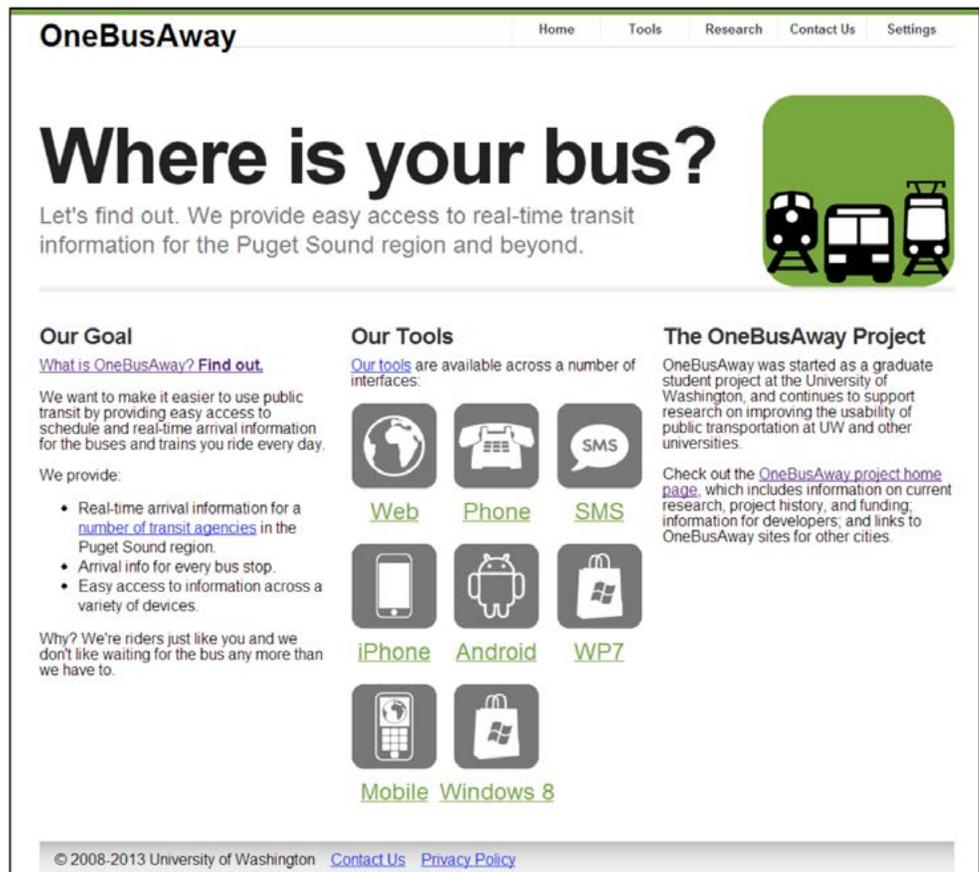
However, offering real-time information services to transit riders has significant challenges. The cost for a transit agency to implement both Automatic Vehicle Location (AVL) technologies and information dissemination technologies (e.g., electronic signs, mobile phone apps) is not trivial, ranging from approximately \$800,000 for a 17-vehicle fleet to \$24 million for a 1,900-vehicle fleet (Parker 2008), especially in the public sector where budgets are under pressure. This estimate does not include the cost of mobile apps, which also is significant. The development cost for a business app that includes real-time information can be upwards of \$150,000 (Lauvray 2011); understandably, agencies have cited development costs as being the primary barrier for offering “official” transit agency mobile apps (Wong et al. 2013). 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 also can 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 [Watkins et al. 2011]), 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 as “open data” (Barbeau 2013; Wong et al. 2013). Third-party developers (individuals not associated with the transit agency) can then independently develop and release mobile apps 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 (San Francisco Bay Area Rapid Transit District 2012), TriMet in Portland (TriMet 2012), Metropolitan Transportation Authority (MTA) in New York (Authority 2012), and Massachusetts Bay Transportation Authority (MBTA) (Massachusetts Bay Transportation Authority 2012). However, these independent developers may not have the same priorities and deadlines as agencies. For example, if a developer does

not 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 U.S. have robust high-tech transit populations and developer communities. In these cities, app growth is more modest (Hillsborough Area Regional Transit 2012). And, since real-time transit data formats often differ between cities, apps for one city cannot be shared easily with another.

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 (University of Washington 2012).

FIGURE 1.
Homepage of open-source
transit information system
OneBusAway in Puget Sound



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 (OneBusAway Organization 2013). In addition to being open-source, OneBusAway supports popular bulk transit data formats such as General Transit Feed Specification (GTFS) (Google 2012), GTFS-realtime (Google 2012), and Service Interface for Real Time Information (SIRI) ((CEN) 2012), which means that anyone with access to transit data in these formats can launch his/her own OneBusAway service for his/her city. Furthermore, OneBusAway includes open-source native mobile apps for iPhone, Android, Windows Phone, and Windows 8, which provide rich functionality and responsiveness beyond that typically available in web applications. OneBusAway has been used to jump-

start several pilot and production deployments of real-time transit information systems (OneBusAway 2012). It also has served as the foundation of several research projects that aim to better understand how real-time information impacts transit riders (Ferris 2010; Ferris et al. 2010; Watkins 2011; Watkins et al. 2011; Gooze et al. 2013; Brakewood 2014).

However, until recently, there was a key limitation with the original OneBusAway project—the OneBusAway mobile apps in the respective app stores (i.e., Google Play, Apple App Store, Windows Phone Store, Windows Store) were configured to work only in Puget Sound, where OneBusAway originally was 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 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 (OneBusAway 2013), 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.

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 PhD dissertation work of Brian Ferris (Ferris 2011) and Kari Watkins (Watkins 2011) and, at the same time, it spread virally to serve 50,000 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-regionally feasible is the growing availability of open transit data and, in particular, the emergence of several *de-facto* transit data standards such as GTFS (Google 2012). As of December 2012, more than 500 agencies worldwide are sharing static (i.e., schedule) data in the GTFS format (Front Seat Management 2012), which allows third-party developers to create transit apps based on these 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 (Roth 2012). While many agencies originally provided GTFS data for Google Transit, many transit and multimodal applications based on GTFS data have emerged (Barbeau and Antrim 2013), 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 (Barbeau 2013). 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 5 minutes.”).

GTFS-realtime and SIRI have emerged as the two most popular fire hose open data formats (Barbeau 2013). 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 also are supported. And since OneBusAway is open-source, support for new formats can be added by any developer (OneBusAway 2012).

As a result, 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 (OneBusAway Organization 2013).

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, Citymapper, and Tiramisu. These “closed-source” applications all are operated by a single entity that has full control over what cities are supported. A city can request to be included, but it may not be added to the service. Business decisions, such as Apple’s choice to remove Google Maps in mid-2012, resulting in the loss of transit directions for iPhone users, also can 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 architecture, 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 previously-mentioned commercially-available solutions.

Multi-Region Architecture

Design Decisions

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

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 for the apps would be changed to 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.”

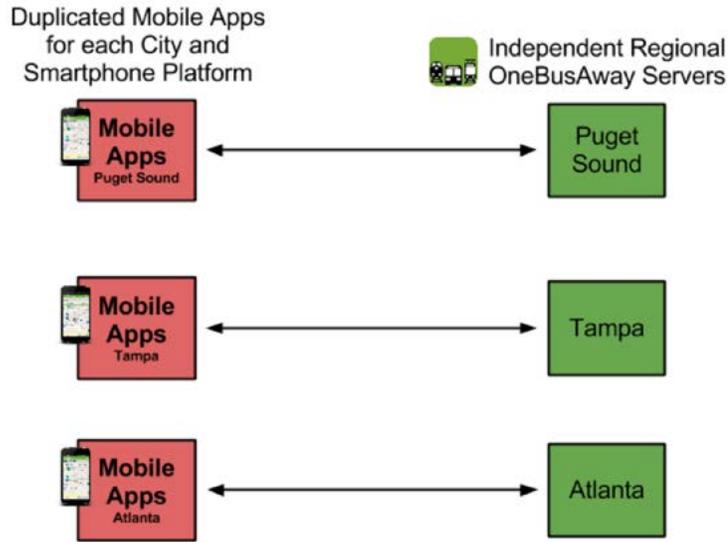
This strategy has the advantage of each city acting independently to deploy mobile apps without requiring any coordination among cities. However, this approach has three major drawbacks:

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.

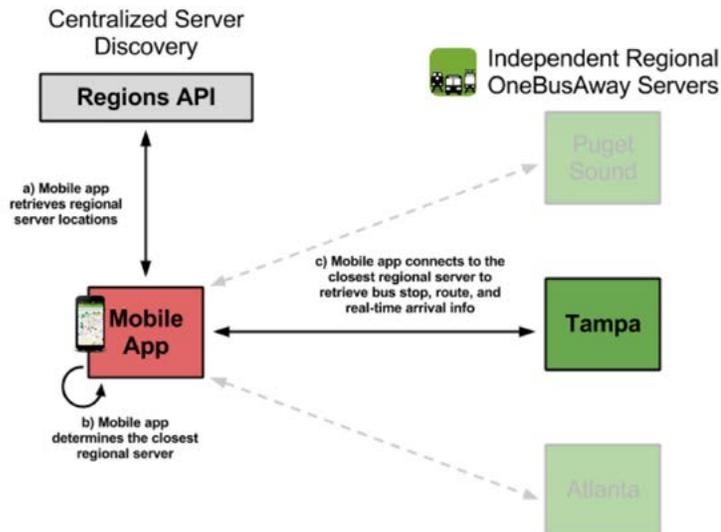
3. *Scalability* – The above two problems increase in complexity as OneBusAway is scaled up to include more and more cities.

FIGURE 2.
Possible design for OneBusAway with duplicated mobile apps for each city



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. 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).

FIGURE 3.
Current OneBusAway multi-region architecture



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 requires 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 and 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 also can 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.

Occasionally, there will be changes to the list of servers and configuration information, including the addition of new regions. Since this information is not expected to change frequently, the mobile app only occasionally refreshes the local copy of region information from the Regions API—once per week in the current design. (This timeframe was selected to balance a reasonable refresh rate to detect new regions against adding additional communication between the mobile device and server, which has an impact on mobile device battery life and increases server load. So far, this timeframe has worked well

in practice.) 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.

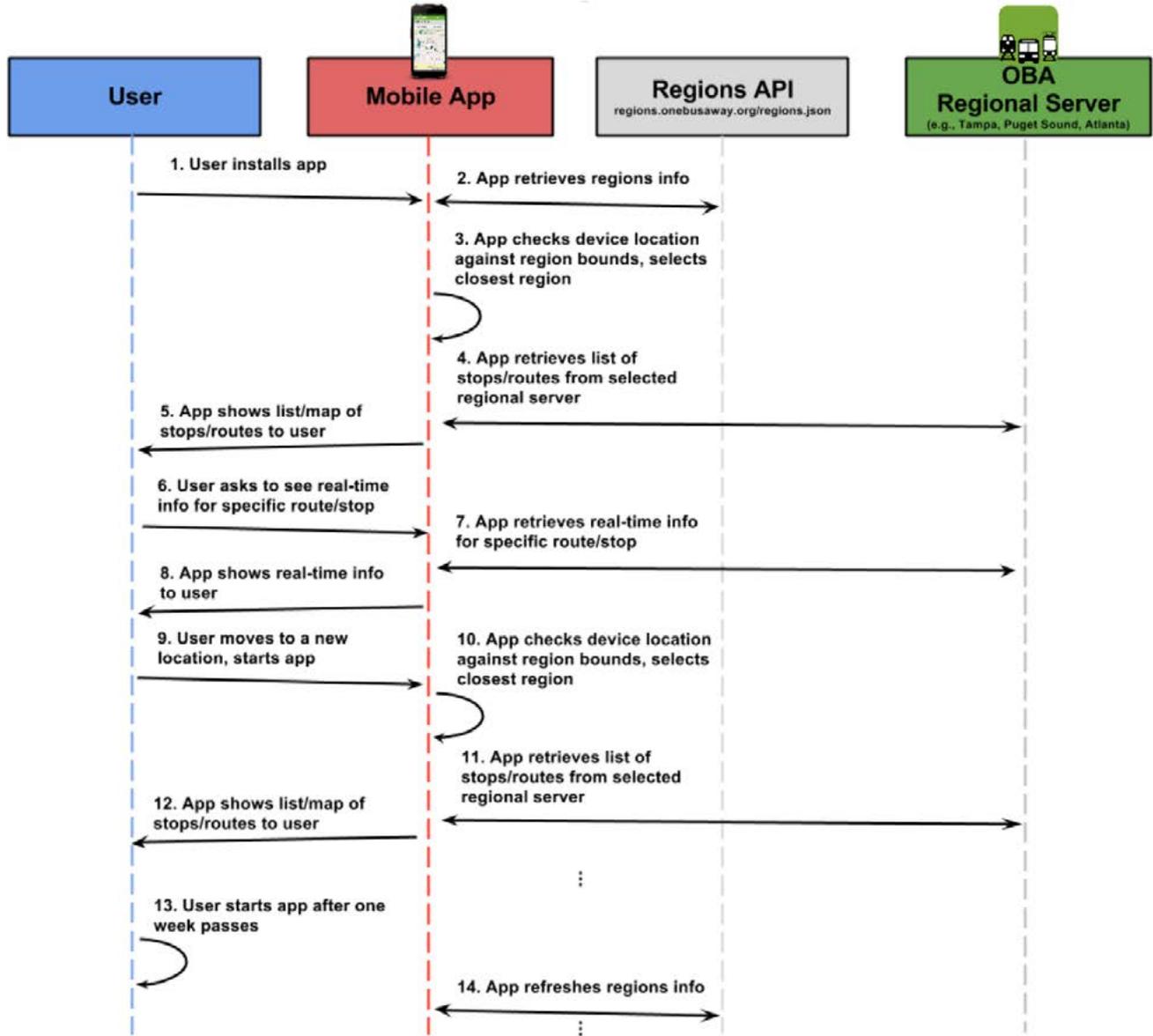


FIGURE 4. Protocol used by mobile apps to connect to a regional OneBusAway server

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 four 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 be resolved 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 “Collaborative Process” section of this paper.

Issue #2 is not difficult to achieve if the third-party developers are actively maintaining their apps and communicating with others participating in the OneBusAway community. If the developer of the app has the development environment set up and another contributor has made the source code modifications, it would take a few hours of effort to review the changes, compile a new release, and publish this new version to the respective app store. 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 25, 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 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 also is 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 handle the majority of feedback and direct any application-specific feedback to the respective application developers as needed. Overall, as discussed later, the OneBusAway mobile app developers were generally enthusiastic participants in this project, since it immediately made their work more widely available to a much larger number of users.

OneBusAway Server Administrators

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 be low effort to implement and maintain, both for the central server directory administrator and 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,” upon any edits. The multi-region administrator runs 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 web server as the Regions API. Thus, adding a new region to the Regions API is fairly simple.

OneBusAway Regions

As of August 2013 (just prior to the launch of the multi-region project), the OneBusAway software suite was deployed to Puget Sound, Tampa, and Atlanta. MTA in New York uses a modified version of OneBusAway for the MTA Bus Time project (Metropolitan Transportation Authority 2012). Detroit has used the OneBusAway software to implement its “Text-My-Bus” text-messaging service for transit riders (Code for America 2012).

In Puget Sound, 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 are provided by a dedicated HTTP server that is made available to OneBusAway, Pierce Transit are provided via FTP from a secure file server, Intercity Transit are provided via HTTP, and Sound Transit data are 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, the 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-real-time feed for Hillsborough Area Regional Transit (HART)’s OrbCAD AVL SQL Server database (University of South Florida 2013) 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 server 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.

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 multi-region architecture design to ensure that the process to add and maintain servers 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 (OneBusAway 2013) served as the primary group email list. The OneBusAway Board of Directors also held scheduled monthly phone calls 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 (OpenPlans 2012), 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 (Software Freedom Conservancy 2013) and Github.com, an online software project hosting site that uses Git for version control. Git is a fully-distributed version control system that allows multiple developers to independently work on a project and then easily merge their contributions back into a single project. Github hosts projects versioned with Git and allows developers to communicate easily via email and the website to discuss issues for fixing bugs or implementing new features. The OneBusAway Github organizational account currently features 39 individual projects, or “code repositories,” and 15 official members are under this account. Among the open-source projects are the main OneBusAway server software and apps for Android, iPhone, Windows Phone, and Windows 8, 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” (create a copy of) any OneBusAway project. These users can then edit and modify the copy 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. Forking a project on Github also provides 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 benefited 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 was completed by a contributor from Tampa who was interested in accelerating the availability of the app in Tampa. The iPhone app had the most contributors (4 in Puget Sound and 1 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. Both the Windows 8 and Windows Phone multi-region updates were completed entirely by the author of the Windows 8 app.

To keep the source code uniform in format and structure, the various OneBusAway projects (e.g., server code, mobile apps) have style guides that can be used by software development tools to re-format any new code to match the project. Additionally, to ensure that the source code remains freely available under a common open-source license, third-party developers are required to sign an Individual Contributor License Agreement (ICLA) that specifies that copyright and patent rights for their contribution are assigned to the project.

Results

In the first half of 2013, the four OneBusAway native mobile applications were modified by mobile app developers to interact with the Regions API as part of the multi-region architecture. In August 2013, the multi-region apps were published on each of the respective app stores and made available in both Atlanta and Tampa, with no perceptible difference to users in Puget Sound. As a result, transit riders in Tampa and Atlanta had 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.

There was substantial growth in the use of OneBusAway in the year following the multi-region launch. In August 2013, the OneBusAway Android app was actively installed on 141,817 devices, with a total of 234,281 downloads for the Puget Sound area. One year later, in August 2014, after launching in Tampa, Atlanta, Washington, DC (beta), York (Canada) (beta), and Bear Transit for the University of California, Berkeley (beta), there were 219,460 active installs with a total of 336,681 downloads. In other words, over the course of one year, more than 77,000 active Android devices were added to the system (approximately 54% increase). iOS users grew by approximately 20 percent (approximately 117,000–140,400) over the same time period. Windows Phone app downloads grew from 41,950 to 60,751, a growth of approximately 44 percent. Windows 8 app use increased by around 3,000 downloads.

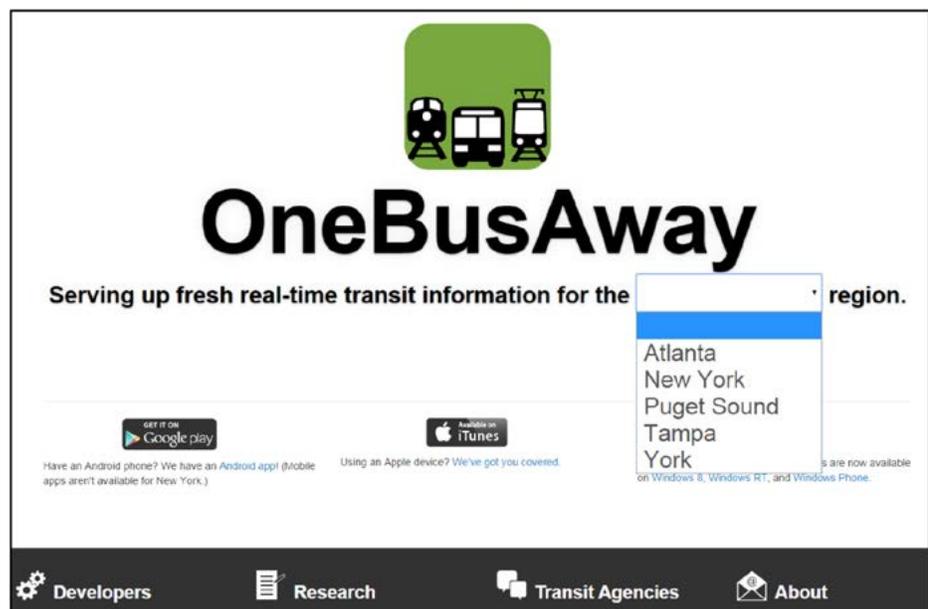
Studies of the effectiveness of OneBusAway regarding the user experience and impacts on transit riders have been reported in multiple papers (Watkins et al. 2011) (Ferris et al. 2010; Ferris 2011; Watkins 2011), including issues with accuracy and rider perception (Gooze et al. 2013). Although these studies took place in Seattle, additional work is being undertaken in Tampa, New York, and Atlanta (Brakewood 2014) (Brakewood et al. 2014). In short, OneBusAway provides an enhanced user experience, especially in regards to the experience of waiting for the bus to arrive.

OneBusAway has proven to be a reliable platform for delivering transit agency data. From August 2013 to August 2014 in Tampa, the only interruptions in service of OneBusAway to customers were related to internal HART networking issues, not problems with the OneBusAway software or hardware infrastructure. To avoid future issues caused by internal network infrastructure, HART moved hosting OneBusAway to a cloud computing service. Since this time, there have been no further interruptions of OneBusAway service to users. As a result, both agencies and riders have generally been pleased with the deployment of OneBusAway. HART Interim Chief Executive Officer Katharine Eagan stated, “We’re excited with how our customers in Tampa have been so quick to use the OneBusAway app. It has truly enhanced the rider’s experience because they have the answers they need right at their fingertips, and it demonstrates that our patrons appreciate our efforts to bring them innovative solutions.”

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, resulting 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 among multiple agencies and universities. New apps continue to emerge as part of this community; in April 2014, a beta version of OneBusAway for Google Glass was created and is available in all OneBusAway regions.

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.
New OneBusAway multi-region website showing multiple cities with shared 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 also are 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.

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.

Over the year following the multi-region launch, other areas expressed interest in being added as new OneBusAway regions, including Washington, DC, York (Canada), and Bear Transit in California. However, these regions had not fully tested their real-time information, nor did they have real-time information available for the all agencies included in the region. As a result, they were not ready for a production launch of OneBusAway, but they did want to test OneBusAway with a small user group. To facilitate this “beta” testing, a new “Experimental” field was added to the Regions API, which is set to “true” for any region that has not yet officially launched (e.g., Washington, DC, Bear Transit). The iPhone and Android apps also were modified to include a new user setting to enable “experimental regions” so that users can easily test new regions. When a region is ready to officially launch OneBusAway, this experimental field is set to “true” and then is visible in the apps by default. Additional details about the differences between experimental and production regions can be found in “Adding Regions to the OneBusAway Multi-Region Scheme” (OneBusAway 2014). York Region Transit/VIVA in Canada went through the beta testing process and was promoted to a full production region in September 2014.

Some agencies have expressed an interest in being able to brand OneBusAway with their own colors or even going so far as deploying new versions of the OneBusAway apps to the app stores that are listed under their agency name. Future work can focus on technical solutions to these problems that would allow agencies to re-brand OneBusAway or at least show their identity within the apps while maintaining a single copy of the source code.

The design of directing email feedback from within the app to the local regional maintainers instead of the app developers has been relatively successful to date. Despite significant growth in the number of users over the last year, only 17 email requests for support were received by the OneBusAway Android application developers. Additionally, very few emails were mistakenly sent to the local region (fewer than 10 for Tampa) rather than to the Android app developers. This design successfully ensures that the app developers will not be overwhelmed as new cities are added to OneBusAway. If support emails are sent

to the incorrect location, they are simply forwarded (e.g., from the region support email to the Android app developer email) as needed.

The vast majority of issues reported via email to HART in Tampa was related to arrival time prediction data quality (e.g., the bus said it would arrive in 5 minutes, but arrived earlier than that). Since data (both schedule and real-time) are provided by the agency, the agency is solely responsible for fixing issues related to data quality. Other transit apps using the same data also would be affected by these issues. The next most popular feedback topic was customer experiences with bus drivers, both negative and positive. Future work could help organize the wealth of information coming from riders back to the agencies to facilitate taking action based on these data.

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 who had not previously contributed to OneBusAway. However, third-party developers can also be unpredictable. During this effort several of the volunteer iPhone developers started and stopped work on the app, primarily due to time pressures from their full-time paid employment. 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 beneficial 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. However, 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 can 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 also is important, in particular for the OneBusAway iPhone app, which is the plat-

form of choice for many visually impaired riders who use it with “VoiceOver” mode. When new features are introduced, it is important to ensure that the app remains accessible to these riders.

Conclusions and Future Work

The OneBusAway multi-region project has succeeded in rapidly expanding mobile apps for Android, iPhone, Windows Phone, and Windows 8 to many new cities outside of the original Puget Sound deployment, including production launches in Tampa, Atlanta, and York. OneBusAway multi-region enables the rapid deployment of these apps to any city, with several more already on the horizon.

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, when writing procurement contracts for OneBusAway installations, software extensions, or maintenance agreements, agencies 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 also will include some funds to support shared resources, such as project-wide software engineers. The role of vendors in the open-source ecosystem also should be examined to ensure sufficient incentives for vendor support of OneBusAway deployments.

Finally, OneBusAway was built on the cornerstone of research about the impacts of real-time information, and the project team continues to improve the functionality and usability of the applications. Multiple research studies regarding ridership impacts are ongoing, including a study about the cost-benefit of providing real-time information via such applications.

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 and 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 they continue to contribute. 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 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-real-time feed in Tampa. The authors would also like to acknowledge their funding sources: 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 who have provided data and resources, including Sound Transit, King County Metro, Pierce Transit, Hillsborough Area Regional Transit, and Metropolitan Atlanta Rapid Transit Authority.

References

- (CEN), E. C. f. S. 2012. CEN/TS 15531 (prCEN/TS-OO278181) SIRI Home Page. Retrieved August 1, 2012, 2012, from <http://www.kizoom.com/standards/siri/>.
- Barbeau, S. 2013. Open transit data—A developer’s perspective. APTA TransITech 2013, Phoenix, Arizona.
- Barbeau, S. J., and A. Antrim. 2013. The many uses of GTFS Data – Opening the door to transit and multimodal applications. ITS America 2013, Nashville, Tennessee.
- Brakewood, C. 2014. Quantifying the impact of real-time information on transit ridership. PhD dissertation, Georgia Institute of Technology.
- Brakewood, C., S. Barbeau, et al. 2014. An experiment evaluating the impacts of real-time transit information on bus riders in Tampa, Florida. *Transportation Research Part A: Policy and Practice*, 69: 409-422.
- Cluett, C., S. Bregman, et al. 2003. Customer preferences for transit ATIS. Federal Transit Administration.
- Code for America. 2012. Text my bus: Transit info for Detroiters. Retrieved July 26, 2013, 2013, from <http://codeforamerica.org/2012/09/11/text-my-bus-transit-info-for-detroiters/>.
- Ferris, B. 2010. Location-aware tools for improving public transit usability. *IEEE Pervasive Computing*, K. Watkins and A. Borning, 9: 13-19.
- Ferris, B. 2011. OneBusAway: Improving the usability of public transit. PhD dissertation, University of Washington.
- Ferris, B., K. Watkins, et al. 2010. OneBusAway: Results from providing real-time arrival information for public transit. *Proceedings of the 28th International Conference on Human Factors in Computing Systems*. Atlanta, ACM: 1807-1816.
- Front Seat Management, LLC. 2012. City-Go-Round. Retrieved March 1, 2012, 2012, from <http://www.citygoround.org/agencies/>.
- Google, Inc. 2012. General Transit Feed Specification reference. Retrieved February 24, 2012, 2012, from <http://developers.google.com/transit/gtfs/reference>.
- Google, Inc. 2012. What is GTFS-realtime? Retrieved August 1, 2012, 2012, from <https://developers.google.com/transit/gtfs-realtime/>.
- Gooze, A., K. Watkins, et al. 2013. Benefits of real-time information and the impacts of data accuracy on the rider experience. Transportation Research Board 92nd Annual Meeting, Washington, DC.

- Hillsborough Area Regional Transit. 2012. App Center. Retrieved November 19, 2012, 2012, from <http://www.gohart.org/developers/appcenter.html>.
- Lauvray, L. 2011. iPad app development cost—A breakdown. Propelics blog.
- Massachusetts Bay Transportation Authority. 2012. App Center. Retrieved November 19, 2012, 2012, from http://www.mbta.com/rider_tools/apps/.
- Metropolitan Transportation Authority. 2012. MTA Bus Time(R). Retrieved August 1, 2012, 2012, from <http://bustime.mta.info/>.
- Metropolitan Transportation Authority. 2012. App Center. Retrieved October 30, 2014, from <http://www.mta.info/apps/>.
- OneBusAway. 2012. OneBusAway deployments. Retrieved August 1, 2012, 2012, from <https://github.com/OneBusAway/onebusaway/wiki/OneBusAway-Deployments>.
- OneBusAway. 2012. Real-time configuration guide. Retrieved August 1, 2012, 2012, from <https://github.com/OneBusAway/onebusaway-application-modules/wiki/Real-Time-Data-Configuration-Guide>.
- OneBusAway. 2013. OneBusAway Developers Group. Retrieved July 26, 2013, 2013, from <https://groups.google.com/forum/#!forum/onebusaway-developers>.
- OneBusAway. 2013. OneBusAway multi-region. Retrieved July 30, 2013, 2013, from <https://github.com/OneBusAway/onebusaway/wiki/Multi-Region>.
- OneBusAway. 2014. Adding regions to the OneBusAway multi-region scheme. Retrieved August 4, 2014, 2014, from <http://onebusaway.org/transit-agencies/adding-regions.php>.
- OneBusAway Organization. 2013. OneBusAway Github source code repository. Retrieved July 16, 2013, 2013, from <https://github.com/OneBusAway/onebusaway-application-modules/wiki>.
- OpenPlans. 2012. OpenTripPlanner. Retrieved August 1, 2012, 2012, from <http://opentripplanner.com/>.
- Parker, D. 2008. TCRP Synthesis 73, AVL systems for bus transit: Update. Transportation Research Board, Washington, DC.
- Roth, M. 2012. How Google and Portland's TriMet set the standard for Open Transit Data. <http://sf.streetsblog.org>.
- San Francisco Bay Area Rapid Transit District. 2012. Third party apps. Retrieved August 1, 2012, 2012, from <http://www.bart.gov/schedules/appcenter/>.
- Software Freedom Conservancy. 2013. Git. Retrieved July 26, 2013, from <http://git-scm.com/>.
- Steinfeld, A., and J. Zimmerman. 2010. Interviews with transit riders in San Francisco and Seattle.
- Tang, L., and P. Thakuria. 2012. Ridership effects of real-time bus information system: A case study in the City of Chicago. *Transportation Research Part C: Emerging Technologies* 22(0): 146-161.

- TriMet. 2012. TriMet App Center. Retrieved August 1, 2012, 2012, from <http://trimet.org/apps/>.
- University of South Florida. 2013. HART-GTFS-realtime Generator Github wiki. Retrieved July 26, 2013, 2013, from <https://github.com/CUTR-at-USF/HART-GTFS-realtime-Generator/wiki>.
- University of Washington. 2012. OneBusAway—Where is your bus? Retrieved August 1, 2012, 2012, from <http://onebusaway.org/>.
- Watkins, K. 2011. Using technology to revolutionize public transportation. PhD dissertation, University of Washington.
- Watkins, K. E., B. Ferris, et al. 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* 45(8): 839-848.
- Windmiller, S., T. Hennessey, et al. 2014. Accessibility of communication technology and the rider experience: A case study of St. Louis Metro. *Transportation Research Record*, in press.
- Wong, J., L. Reed, et al. 2013. Open Transit Data: State of the practice and experiences from participating agencies in the United States. Transportation Research Board 92nd Annual Meeting, Washington, DC: 15.
- Zimmerman, J., A. Tomasic, et al. 2011. Field trial of Tiramisu: Crowd-sourcing bus arrival times to spur co-design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, ACM.

About The Authors

DR. SEAN J. BARBEAU (barbeau@cutr.usf.edu) is the Principal Mobile Software Architect for R&D at the Center for Urban Transportation Research at the University of South Florida. His research interests include intelligent location-aware services for mobile phones, lightweight data communication frameworks for mobile devices, mobile application optimization to conserve battery life, spatial data mining and prediction of individualized travel behavior. He has more than 40 peer-reviewed papers and conference presentations and was a member of an international Expert Group that developed a Java-based software standard for location-aware mobile phones. He has 13 patented inventions and 2 patents pending in the area of location-aware technology. He has a BS and an MS in Computer Science and a PhD in Computer Science and Engineering from USF.

DR. ALAN BORNING (borning@cs.washington.edu) is a professor in the Department of Computer Science & Engineering at the University of Washington and an adjunct faculty member in the Information School. His research interests are human-computer interaction and designing for human values and object-oriented and constraint-based programming languages. Current projects include tools for making public transit more usable, systems to support civic engagement and participation, and constraint-based programming languages and systems. He received a BA in Mathematics from Reed College in 1971 and a PhD in Computer Science from Stanford University in 1979. Awards include a Fulbright

Senior Scholar Award for lecturing and research in Australia and being named a Fellow of the Association for Computing Machinery in 2001.

DR. KARI EDISON WATKINS (*kari.watkins@ce.gatech.edu*) is an Assistant Professor in the School of Civil and Environmental Engineering at the Georgia Institute of Technology. She returned to her undergraduate alma mater to become a faculty member in 2011 after completing a PhD in Civil and Environmental Engineering at the University of Washington. Her teaching and research interests revolve around multimodal transportation planning and the use of technology in transportation. Her dissertation about the impacts of transit rider information tools won a Council of University Transportation Centers' Wootan Award, and she recently was recognized by *Mass Transit Magazine* in its "Top 40 under 40."