

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

3
4 Sean J. Barbeau, Ph.D. (Corresponding Author)
5 Center for Urban Transportation Research
6 University of South Florida
7 Tampa, FL 33620
8 813-974-7208
9 barbeau@cutr.usf.edu

10
11 Alan Borning, Ph.D.
12 Department of Computer Science & Engineering
13 University of Washington
14 Seattle, WA 98195
15 206-543-6678
16 borning@cs.washington.edu

17
18 Kari Watkins, Ph.D.
19 School of Civil and Environmental Engineering
20 Georgia Institute of Technology
21 790 Atlantic Drive, Atlanta, GA 30332
22 206-250-4415
23 kari.watkins@ce.gatech.edu

24
25 November 15, 2013

26
27 5,928 words + 5 figures (1,250) = 7,178 words

28 **ABSTRACT**

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

46

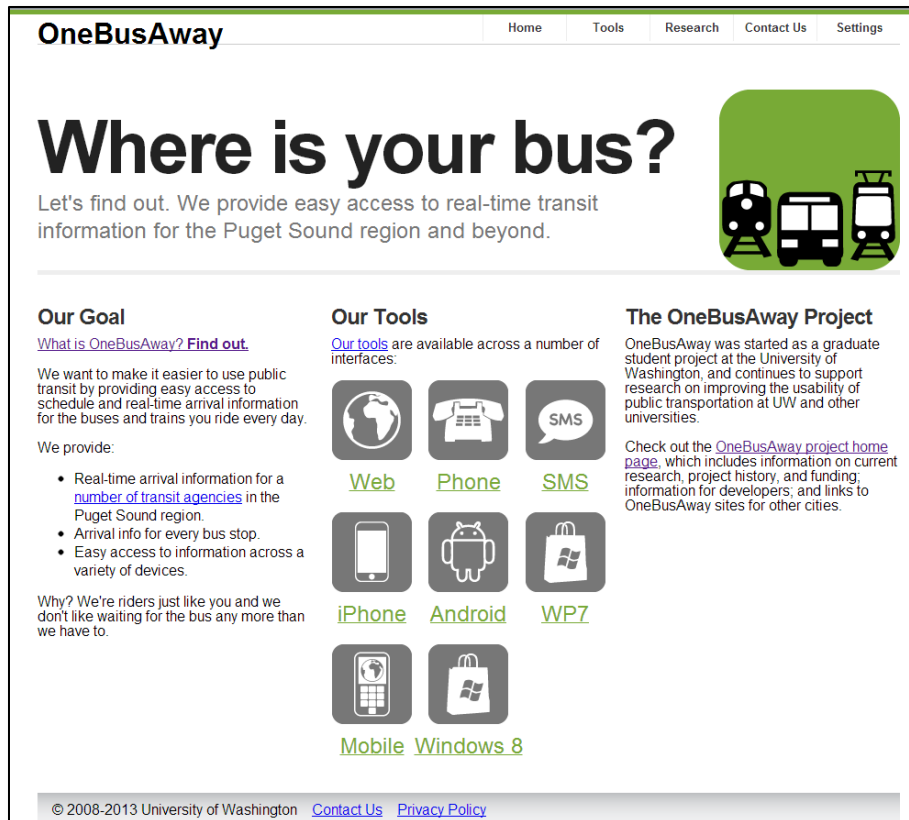
1 1. INTRODUCTION

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

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

30 One strategy for increasing the number of mobile transit apps at a transit agency is for the
31 agency to share static (i.e., schedule) and real-time transit information with the general public [9,
32 10]. Third-party developers (i.e., individuals not associated with the transit agency) can then
33 independently develop and release mobile apps that use this data to the general public. This
34 strategy has successfully produced a number of third-party transit apps at several agencies in the
35 U.S., including Bay Area Regional Transit (BART) in San Francisco [11], TriMet in Portland
36 [12], Metropolitan Transportation Authority (MTA) in New York [13], and Massachusetts Bay
37 Transportation Authority (MBTA) [14].

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



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

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

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

17 However, until recently there was a key limitation with the original OneBusAway project
18 – the OneBusAway mobile apps on the respective app stores (i.e., Google Play, Apple App
19 Store, Windows Phone Store, Windows Store) were only configured to work in Puget Sound,
20 WA, where OneBusAway was originally developed.
21

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

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

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

20 **2. BACKGROUND**

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

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

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

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

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

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

14 **Comparison to other real-time transit applications**

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

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

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

37
38

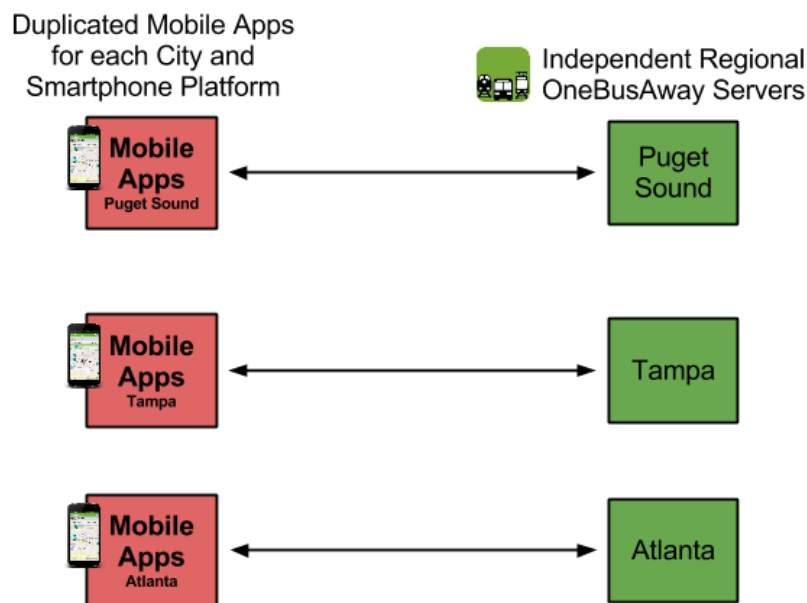
1 3. MULTI-REGION ARCHITECTURE

2 Design decisions

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

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

10



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

12

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

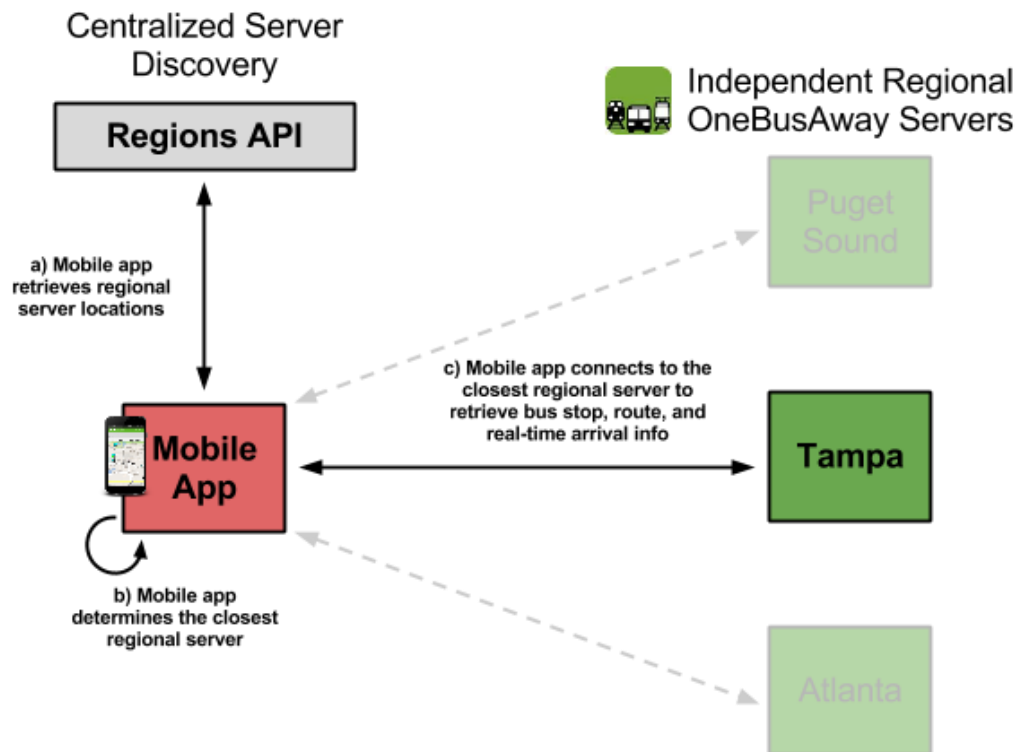
18 This strategy has the advantage of each city acting independently to deploy mobile apps,
19 without requiring any coordination among cities. However, this approach has three major
20 drawbacks:

- 21 1. *Sustainability* – Each city would need to find new developers to maintain and update the
22 local Android, iPhone, Windows Phone, and Windows 8 apps. This is clearly
23 undesirable, as it is already challenging for many cities to find developers interested in
24 developing transit apps.
- 25 2. *Fragmentation* – There would be one copy of each mobile app source code for each city.
26 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

1 that is difficult to maintain, limiting shared app improvements among cities.

2 Additionally, when users try to download the app from the respective app store, they
 3 would be presented with a list of OneBusAway apps from all cities to choose from (e.g.,
 4 “OneBusAway Tampa”, “OneBusAway Atlanta”) which places the burden on the user to
 5 find and install the correct app.

- 6 3. *Scalability* – The above two problems increase in complexity as OneBusAway is scaled
 7 up to include more and more cities.
 8



9
 10
 11 **FIGURE 3 OneBusAway Multi-region Architecture efficiently extends mobile apps to new**
 12 **cities**

13 An alternate approach is for a group of pilot cities to work together and create a
 14 coordinated OneBusAway multi-region system (FIGURE 3). Here, a centralized OneBusAway
 15 directory is created with a list of known OneBusAway servers in various cities. Then, the
 16 existing iPhone, Android, Windows Phone, and Windows 8 apps are modified so they discover
 17 available OneBusAway servers from the directory (i.e., “Regions API”), as shown in FIGURE
 18 3a. Then, the app compares the user’s real-time location to the list of server locations (FIGURE
 19 3b), and then connects to the closest server to retrieve route, stop, and arrival information
 20 (FIGURE 3c).

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

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

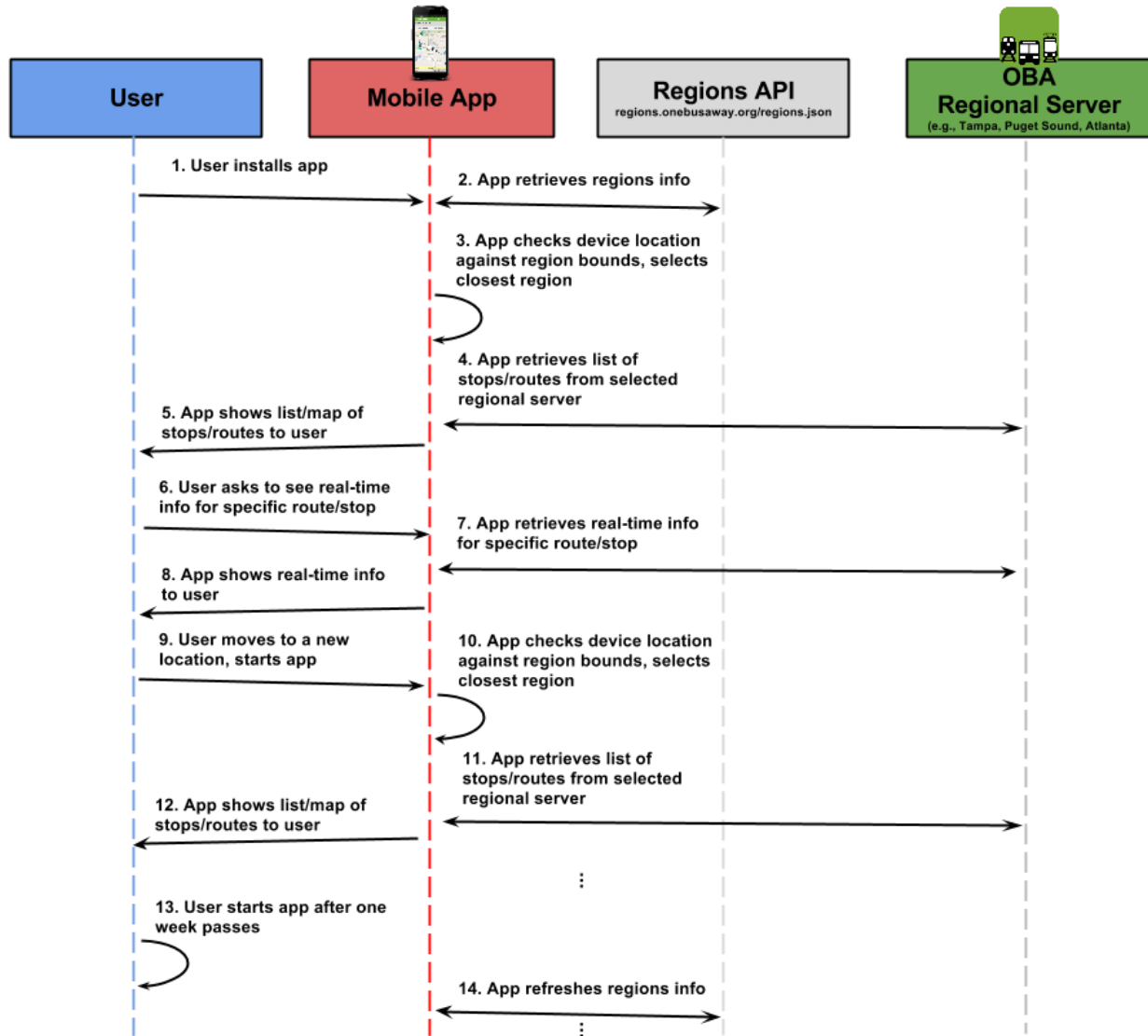
6 **Detailed Protocol**

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

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

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

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



1
2 **FIGURE 4 Protocol used by a mobile app to determine which OneBusAway regional**
3 **server it should connect to, via the Regions API**

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

1 **Mobile App Modifications**

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

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

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

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

27 **OneBusAway Server Administrators**

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

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

41 **OneBusAway Regions**

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

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

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

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

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

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

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

24 **4. COLLABORATIVE PROCESS**

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

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

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

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

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

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

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

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

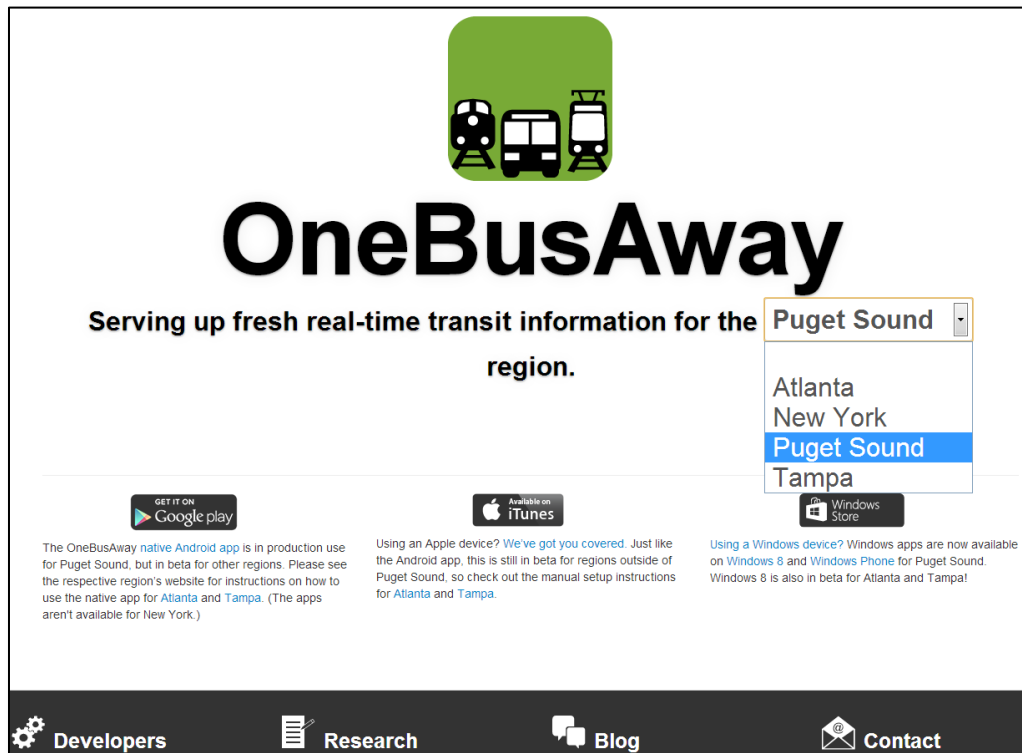
32 5. RESULTS

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

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

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

3 In conjunction with the multi-region app launch, the home page for the OneBusAway
4 project at <http://onebusaway.org> was converted from being specific to Puget Sound to
5 encompassing all cities involved in the project (FIGURE 5).
6
7



8
9

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

13 This allows riders to conveniently access regional OneBusAway services. Information
14 for transit agencies interested in their own OneBusAway deployments, developers who want to
15 contribute to the project, and researchers interested in academic publications related to
16 OneBusAway are also included. A straightforward naming scheme for region URLs (e.g.,
17 <http://tampa.onebusaway.org>, <http://pugetsound.onebusaway.org>) makes it easy to add new
18 regions, while at the same time maintaining the identity of the project as a whole.
19

20 6. LESSONS LEARNED

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

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

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

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

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

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

32 An important consideration for testing is to ensure that the apps are tested on a range of
33 mobile devices (i.e. different models of Android, iPhone, Windows Phone). Accessibility testing
34 is also important, in particular for the OneBusAway iPhone app, which is the platform of choice
35 for many blind riders who use it with "VoiceOver" mode.
36

37 **7. CONCLUSIONS AND FUTURE WORK**

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

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

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

4 As OneBusAway deployments are transferred from universities to transit agencies, it has
5 become evident that procurement best practices should be established. Current recommendations
6 include that agencies, when writing procurement contracts for OneBusAway installations,
7 software extensions, or maintenance agreements, require that any customizations and extensions
8 be open source, and written in a way that they can be contributed back to the project as a whole
9 and benefit all regions, not just the requestor. Ideally any procurement requests will also include
10 some funds to support shared resources, such as project-wide software engineers. The role of
11 vendors in the open-source ecosystem should also be examined to ensure sufficient incentives for
12 vendor support of OneBusAway deployments. Additional recommendations should also be
13 established as new lessons are learned.
14

15 8. ACKNOWLEDGMENTS

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

31 Finally, we thank our supporters at the transit agencies that have provided data and
32 resources, including Sound Transit, King County Metro, Pierce Transit, Hillsborough Area
33 Regional Transit, and Metropolitan Atlanta Rapid Transit Authority.
34

35 9. REFERENCES

- 36 [1] Kari Edison Watkins, Brian Ferris, Alan Borning, G. Scott Rutherford, and David Layton
37 (2011), "Where Is My Bus? Impact of mobile real-time information on the perceived and
38 actual wait time of transit riders," *Transportation Research Part A: Policy and Practice*,
39 Vol. 45 pp. 839-848.
- 40 [2] C. Cluett, S. Bregman, and J. Richman (2003). "Customer Preferences for Transit ATIS,"
41 Federal Transit Administration. Available at
42 http://ntl.bts.gov/lib/jpodocs/repts_te/13935/13935.pdf#sthash.jwn5Oltr.dpuf
- 43 [3] Lei Tang and Piyushimita Thakuriah (2012), "Ridership effects of real-time bus
44 information system: A case study in the City of Chicago," *Transportation Research Part*
45 *C: Emerging Technologies*, Vol. 22 pp. 146-161.

- 1 [4] Aaron Steinfeld and John Zimmerman, "Interviews with transit riders in San Francisco
2 and Seattle," ed, 2010.
- 3 [5] John Zimmerman, Anthony Tomasic, Charles Garrod, Daisy Yoo, Chaya
4 Hiruncharoenvate, Rafae Aziz, Nikhil Ravi Thiruvengadam, Yun Huang, and Aaron
5 Steinfeld (2011), "Field trial of Tiramisu: crowd-sourcing bus arrival times to spur co-
6 design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing*
7 *Systems*, Vancouver, BC, Canada, pp. 1677-1686,
- 8 [6] Brian Ferris, Kari Watkins, and Alan Borning (2010), "OneBusAway: results from
9 providing real-time arrival information for public transit," in *Proceedings of the 28th*
10 *International CHI Conference on Human Factors in Computing Systems*, Atlanta,
11 Georgia, USA, pp. 1807-1816,
- 12 [7] A. Gooze, K. Watkins, and A. Borning (2013), "Benefits of Real-Time Information and
13 the Impacts of Data Accuracy on the Rider Experience," in *Transportation Research*
14 *Board 92nd Annual Meeting*, Washington, D.C., January 13, 2013.
- 15 [8] Larry Lauvray "iPad App Development Cost - A Breakdown." *Propelics Blog*, Vol.
16 Accessed at: <http://www.propelics.com/ipad-app-development-cost-a-breakdown/>
- 17 [9] James Wong, Landon Reed, Kari Watkins, and Regan Hammond, "Open Transit Data:
18 State of the Practice and Experiences from Participating Agencies in the United States,"
19 presented at the Transportation Research Board 92nd Annual Meeting, Washington,
20 D.C., USA, 2013.
- 21 [10] S. Barbeau (2013), "Open Transit Data – A Developer’s Perspective," in *APTA*
22 *TransITech 2013*, Phoenix, Arizona, March 20th, 2013. Accessed at:
23 [http://www.slideshare.net/sjbarbeau/apta-transitech-2013-open-transit-data-a-developers-](http://www.slideshare.net/sjbarbeau/apta-transitech-2013-open-transit-data-a-developers-perspective)
24 [perspective](http://www.slideshare.net/sjbarbeau/apta-transitech-2013-open-transit-data-a-developers-perspective)
- 25 [11] San Francisco Bay Area Rapid Transit District. "Third Party Apps." Accessed August 1,
26 2012 from <http://www.bart.gov/schedules/appcenter/>
- 27 [12] TriMet. "TriMet App Center." Accessed August 1, 2012 from <http://trimet.org/apps/>
- 28 [13] Metropolitan Transportation Authority. "App Center." Accessed November 19, 2012
29 from <http://www.mta.info/apps/>
- 30 [14] Massachusetts Bay Transportation Authority. "App Center." Accessed November 19,
31 2012 from http://www.mbta.com/rider_tools/apps/
- 32 [15] Hillsborough Area Regional Transit. "App Center." Accessed November 19, 2012 from
33 <http://www.gohart.org/developers/appcenter.html>
- 34 [16] University of Washington. "OneBusAway - Where is your bus?". Accessed August 1,
35 2012 from <http://onebusaway.org/>
- 36 [17] OneBusAway Organization. "OneBusAway Github Source Code Repository." Accessed
37 July 16, 2013 from [https://github.com/OneBusAway/onebusaway-application-](https://github.com/OneBusAway/onebusaway-application-modules/wiki)
38 [modules/wiki](https://github.com/OneBusAway/onebusaway-application-modules/wiki)
- 39 [18] Google, Inc. "General Transit Feed Specification Reference." Accessed February 24,
40 2012 from <http://developers.google.com/transit/gtfs/reference>
- 41 [19] Google, Inc. "What is GTFS-realtime?". Accessed August 1, 2012 from
42 <https://developers.google.com/transit/gtfs-realtime/>
- 43 [20] European Committee for Standardization (CEN). "CEN/TS 15531 (prCEN/TS-
44 OO278181) SIRI Home Page." Accessed August 1, 2012 from
45 <http://www.kizoom.com/standards/siri/>

- 1 [21] OneBusAway. "OneBusAway Deployments." Accessed August 1, 2012 from
2 <https://github.com/OneBusAway/onebusaway/wiki/OneBusAway-Deployments>
- 3 [22] OneBusAway. "OneBusAway Multi-region." Accessed July 30, 2013 from
4 <https://github.com/OneBusAway/onebusaway/wiki/Multi-Region>
- 5 [23] Brian Ferris (2011), "OneBusAway: Improving the Usability of Public Transit," Doctor
6 of Philosophy Thesis, Computer Science and Engineering, University of Washington.
- 7 [24] K. Watkins (2011), "Using Technology to Revolutionize Public Transportation," Doctor
8 of Philosophy Thesis, Civil and Environmental Engineering, University of Washington.
- 9 [25] Front Seat Management, LLC. "City-Go-Round." Accessed March 1, 2012 from
10 <http://www.citygoround.org/agencies/>
- 11 [26] Matthew Roth. (2012). "How Google and Portland's TriMet Set the Standard for Open
12 Transit Data." *SF.STREETSBLOG.ORG*. January 5, 2010. Accessed: from
13 [http://sf.streetsblog.org/2010/01/05/how-google-and-portlands-trimet-set-the-standard-](http://sf.streetsblog.org/2010/01/05/how-google-and-portlands-trimet-set-the-standard-for-open-transit-data/)
14 [for-open-transit-data/](http://sf.streetsblog.org/2010/01/05/how-google-and-portlands-trimet-set-the-standard-for-open-transit-data/)
- 15 [27] Sean J. Barbeau and Aaron Antrim (2013), "The Many Uses of GTFS Data – Opening the
16 Door to Transit and Multimodal Applications," in *ITS America 2013*, Nashville,
17 Tennessee, April 22, 2013. Accessed at: [http://prezi.com/-69luw8sfabp/the-many-uses-](http://prezi.com/-69luw8sfabp/the-many-uses-of-gtfs-data-its-america-april-2013/)
18 [of-gtfs-data-its-america-april-2013/](http://prezi.com/-69luw8sfabp/the-many-uses-of-gtfs-data-its-america-april-2013/)
- 19 [28] OneBusAway. "Real-Time Configuration Guide." Accessed August 1, 2012 from
20 [https://github.com/OneBusAway/onebusaway-application-modules/wiki/Real-Time-](https://github.com/OneBusAway/onebusaway-application-modules/wiki/Real-Time-Data-Configuration-Guide)
21 [Data-Configuration-Guide](https://github.com/OneBusAway/onebusaway-application-modules/wiki/Real-Time-Data-Configuration-Guide)
- 22 [29] Metropolitan Transportation Authority. "MTA Bus Time(R)." Accessed August 1, 2012
23 from <http://bustime.mta.info/>
- 24 [30] Code for America. "Text My Bus: Transit Info for Detroiters." Accessed July 26, 2013
25 from <http://codeforamerica.org/2012/09/11/text-my-bus-transit-info-for-detroiters/>
- 26 [31] University of South Florida. "HART-GTFS-realtimeGenerator Github wiki." Accessed
27 July 26, 2013 from [https://github.com/CUTR-at-USF/HART-GTFS-](https://github.com/CUTR-at-USF/HART-GTFS-realtimeGenerator/wiki)
28 [realtimeGenerator/wiki](https://github.com/CUTR-at-USF/HART-GTFS-realtimeGenerator/wiki)
- 29 [32] OneBusAway. "OneBusAway Developers Group." Accessed July 26, 2013 from
30 <https://groups.google.com/forum/#!forum/onebusaway-developers>
- 31 [33] OpenPlans. "OpenTripPlanner." Accessed August 1, 2012 from
32 <http://opentripplanner.com/>
- 33 [34] Software Freedom Conservancy. "Git." Accessed July 26, 2013 from <http://git-scm.com/>
- 34
- 35