NITC
NATIONAL INSTITUTE for TRANSPORTATION and COMMUNITIES

FINAL REPORT

Rapidly Expanding Mobile Apps for Crowd-sourcing Bike Data to New Cities

NITCN-RR-857 March 2017

NITC is a U.S. Department of Transportation national university transportation center.

TREC
Transportation Insight for Vibrant Communities

RAPIDLY EXPANDING MOBILE APPS FOR CROWD-SOURCING BIKE DATA TO NEW CITIES

Final Report

NITCN-RR-857

by

Dr. Sean J. Barbeau
Center for Urban Transportation Research
University of South Florida

for

National Institute for Transportation and Communities (NITC)
P.O. Box 751
Portland, OR 97207

March 2017

Technical Report Documentation Page

1. Report No. NITCN-RR-857	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Rapidly Expanding Mobile Apps for Crowd-sourcing Bike Data to New Cities	5. Report Date March 2017 6. Performing Organization Code
7. Author(s) Dr. Sean J. Barbeau, Cagri Cetin	8. Performing Organization Report No.
9. Performing Organization Name and Address University of South Florida 4202 E. Fowler Ave. CUT100 Tampa, FL 33620	10. Work Unit No. (TRAIS) 11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Institute for Transportation and Communities (NITC) P.O. Box 751 Portland, Oregon 97207	13. Type of Report and Period Covered 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Cities such as San Francisco, Atlanta, and Portland are using novel methods of data collection to learn more about the use of their bicycle infrastructure. These data can help transportation planners better design or upgrade bicycle facilities. San Francisco created an open-source project, Cycle Tracks, a mobile app used to collect bike path data from bicyclists' smartphones. These data then were used in the SF-CHAMP travel demand model to forecast how attractive Bike Facility A would be compared to Bike Facility B to understand the potential mode shifts that could occur with implementation of bike infrastructure, and to better understand the impact of new SF bike infrastructure on bicyclist travel behavior. A similar project, Cycle Atlanta, was implemented in Atlanta, GA, and was based on the Cycle Tracks open-source code, as was ORcycle for Portland, OR. These methods of gathering data from the public via mobile apps are referred to as "crowd-sourcing."

Whereas open-source crowd-sourcing mobile apps can provide a wealth of information to transportation planners, there is at least one major obstacle to deploying these projects in new cities: software engineers for iOS and Android must modify and re-deploy these apps for each city. As a result, deploying these apps in new cities can be very costly, which limits adoption and removes opportunities for innovation based on data collected from such apps.

This project takes the first steps towards helping to overcome the barriers to wide-scale adoption of bike data crowd-sourcing mobile apps by creating a proof-of-concept "multi-region" architecture, allowing cities to share the same set of mobile apps on the Android and iOS apps stores while setting up their own server specific to their geographic area. This solution can reduce the cost of deployment by leveraging the mobile apps that already exist, rather than each city needing to modify and launch its own version of the apps. Future work should focus on developing a brand identity of these multi-region apps, including a possible partnership with existing organizations that have deployed existing apps (e.g., Georgia Tech for Cycle Atlanta, San Francisco County Transportation Authority for Cycle Tracks) to further test and release the multi-region improvements as updates to their existing applications. Use of the standardized Open311 format to report issues can also be examined, along with leveraging other open-source transportation applications such as OneBusAway to give users additional incentive and value to continue to contribute trip data.

17. Key Words Mobile app, bicycle, data collection, crowd-sourcing, gps, location	18. Distribution Statement No restrictions. Copies available from NITC: www.nitc.us

19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 29	22. Price

ACKNOWLEDGEMENTS

The PI would like to acknowledge Georgia Tech's Cycle Atlanta open-source project (http://CycleAtlanta.org/), and the San Francisco County Transportation Authority's Cycle Tracks project (http://www.sfcta.org/modeling-and-travel-forecasting/cycletracks-iphone-and-android), which were used as the foundation for the project described in this report. Source code from the OneBusAway Open-Source Project, specifically the iOS and Android applications, was also used under the Apache v2.0 license to help implement the multi-region features for this project.

DISCLAIMER

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Cities such as San Francisco, Atlanta, and Portland are using novel methods of data collection to learn more about the use of their bicycle infrastructure. These data can help transportation planners better design or upgrade bicycle facilities. San Francisco created an open-source project, Cycle Tracks (San Francisco County Transportation Authority, 2015), a mobile app used to collect bike path data from bicyclists' smartphones. These data then were used in the SF-CHAMP travel demand model to forecast how attractive Bike Facility A would be compared to Bike Facility B to understand the potential mode shifts that could occur with implementation of bike infrastructure, and to better understand the impact of new SF bike infrastructure on bicyclist travel behavior. A similar project, Cycle Atlanta (Cycle Atlanta, 2015), was implemented in Atlanta, GA, and was based on the Cycle Tracks open-source code, as was ORcycle for Portland, OR (Portland State University, 2015). These methods of gathering data from the public via mobile apps are referred to as "crowd-sourcing."

Whereas open-source crowd-sourcing mobile apps can provide a wealth of information to transportation planners, there is at least one major obstacle to deploying these projects in new cities: software engineers for iOS and Android must modify and re-deploy these apps for each city. As a result, deploying these apps in new cities can be costly, which limits adoption and removes opportunities for innovation based on data collected from such apps.

This project takes the first steps towards helping to overcome the barriers to wide-scale adoption of bike data crowd-sourcing mobile apps by creating a proof-of-concept "multi-region" architecture that allows cities to share the same set of mobile apps on the respective apps stores while setting up their own server specific to their geographic area. This solution can reduce the cost of deployment by leveraging the mobile apps that already exist, rather than each city needing to modify and launch its own version of the app.

Future work should focus on developing a brand identity of these multi-region apps, including a possible partnership with existing organizations that have deployed existing apps (e.g., Georgia Tech for Cycle Atlanta, San Francisco County Transportation Authority (SFCTA) for Cycle Tracks) to further test and release the multi-region improvements as updates to their existing applications. A future phase of this project can choose sample cities to deploy the multi-region apps to in order to collect real data from actual bicyclists. The Open311 standard (Open311, 2016) should also be investigated as a possible format for submitting assets and notes data collected by the Cycle Atlanta multi-region apps, which would allow cities to leverage their existing issue management tools compatible with Open311. Finally, future work could also examine integrating the multi-region bike data recording functionality into other applications that provide more ongoing value to the user. For example, data recording functionality could be integrated into OneBusAway, which currently provides real-time transit arrival information. If additional bike-related functionality is added to OneBusAway to encourage the use by bicyclists (e.g. real-time bike-share information), bike trip data could be recorded via that app, potentially without requiring ongoing user interaction. While passive data recording is much less burdensome to the user, it also does not directly collect more detailed information such as trip purpose or mode of transportation (assuming multimodal travel). Past techniques have examined extracting trip purpose and mode of transportation using automated data mining techniques, which could also be used. A less intrusive minimalist survey technique, where the user is intelligently prompted for small pieces of information at opportune times (e.g., asking the user how they got to a bus stop when they are waiting for a bus), could also be examined.

1.0 INTRODUCTION

1.1 BACKGROUND

Cities such as San Francisco, Atlanta, and Portland are using novel methods of data collection to learn more about the use of their bicycle infrastructure. These data can help transportation planners better design or upgrade bicycle facilities. San Francisco created an open-source project, Cycle Tracks (San Francisco County Transportation Authority, 2015), a mobile app used to collect bike path data from bicyclists' smartphones. These data then were used in the SF-CHAMP travel demand model to forecast how attractive Bike Facility A would be compared to Bike Facility B to understand the potential mode shifts that could occur with implementation of bike infrastructure, and to better understand the impact of new SF bike infrastructure on bicyclist travel behavior. A similar project, Cycle Atlanta (Cycle Atlanta, 2015), was implemented in Atlanta, GA, and was based on the Cycle Tracks open-source code, as was ORcycle for Portland, OR (Portland State University, 2015). These methods of gathering data from the public via mobile apps are referred to as "crowd-sourcing."

Bike data crowd-sourcing mobile apps deployed by cities have some advantages over other bicyclist GPS data sources, such as datasets sold by Strava (Watkins et al., 2016). Watkins et al. found significant demographic differences between Strava users and Cycle Atlanta users. For example, the Strava data contained higher median speeds than Cycle Atlanta, with more trips in the late and very early hours, even in commute trips. Cycle Atlanta users exhibited a strong preference for bike paths, cycle tracks, and low-speed roads. Watkins et al. concluded that Strava users are more experienced "spandex-clad" cyclists willing to travel in conditions that less experienced cyclists may avoid. Additionally, while both Strava and Cycle Atlanta data can be used for gap analysis to understand roads avoided by cyclists, only apps like Cycle Atlanta can provide the individual path data needed for route choice models or other cyclist behaviors, particularly in regards to segmenting data by user characteristics. This is because Strava data contains only bike trip counts on road segments, but, unlike Cycle Atlanta, it does not contain a full per-trip path from a user's origin to destination. Another emerging app that collects trip data from cyclists is Ride Report (Ride Report, 2016). However, Ride Report is a commercial (i.e., not open-source) app and the company only offers data to cities that purchase the information – it may not be accessible to others such as researchers or the general public. Additionally, the raw data for origin/destination paths for each rider, needed for route choice models, is not available via Ride Report.

Whereas open-source crowd-sourcing mobile apps like Cycle Tracks and Cycle Atlanta can provide a wealth of information to transportation planners, there is at least one major obstacle to deploying these projects in new cities: software engineers for iOS and Android must modify and re-deploy these apps for each city, as shown in Figure 1.1. As a result, deploying these apps in new cities can be very costly, which limits adoption and removes opportunities for innovation based on data collected from such apps.

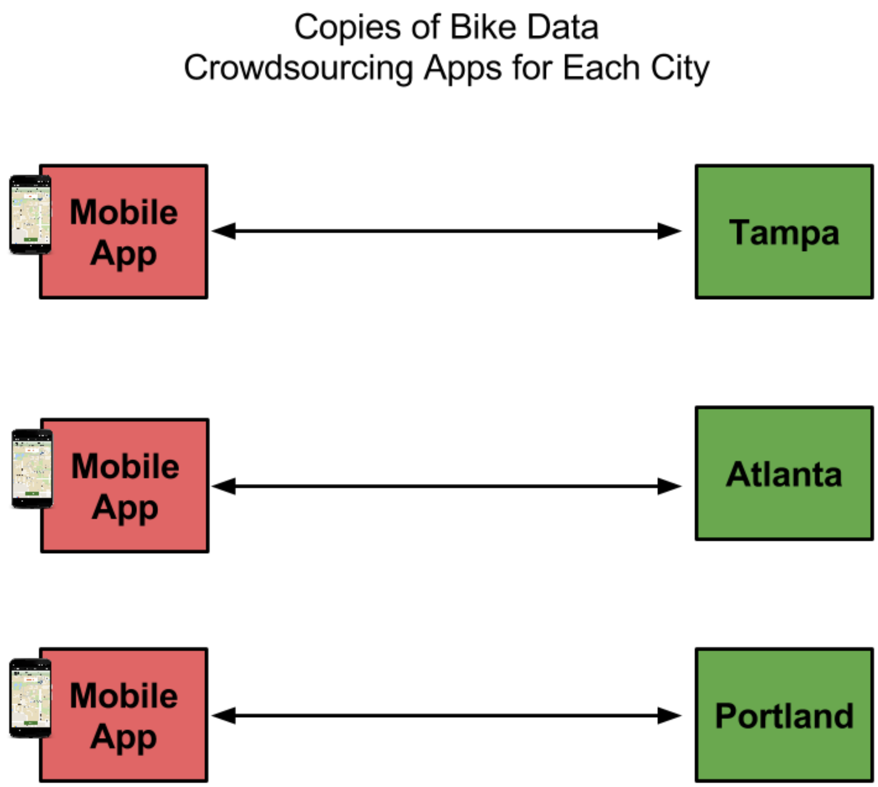


Figure 1.1: The single-region design used by existing apps

This project takes the initial steps towards helping to overcome the barriers to wide-scale adoption of bike data crowd-sourcing mobile apps by creating a proof-of-concept “multi-region” architecture, allowing cities to share the same set of mobile apps on the respective apps stores while setting up their own server specific to their geographic area. This solution reduces the cost of deployment by leveraging the mobile apps that already exist, rather than each city needing to modify and launch its own version of the apps. This multi-region model has been successfully used with the OneBusAway Open-Source Project (http://onebusaway.org/), which has grown to over 10 regions in the past four years (Barbeau, 2014).

1.2 EXISTING APPS

Two existing open-source crowd-sourcing bike applications, Cycle Tracks (http://www.sfcta.org/modeling-and-travel-forecasting/Cycle Tracks-iphone-and-android) and Cycle Atlanta (http://Cycle Atlanta.org/), were examined in this project. Other applications such as Portland’s ORcycle also exist, but the source code for that application was not available at the time of this project. Both Cycle Tracks and Cycle Atlanta applications record user bicycle trip routes and times.

The most recent software development and updates to the Cycle Tracks app were in 2010, therefore reflecting an older Android design. Figures 1.2 and 1.3 are screenshots from Cycle Tracks.

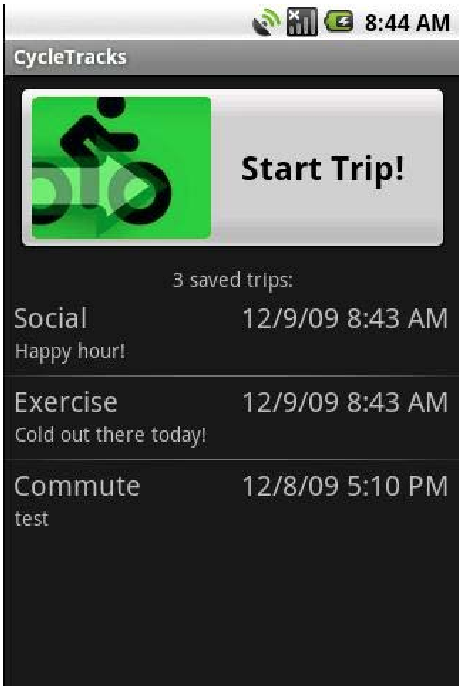


Figure 1.2: Cycle Tracks main screen

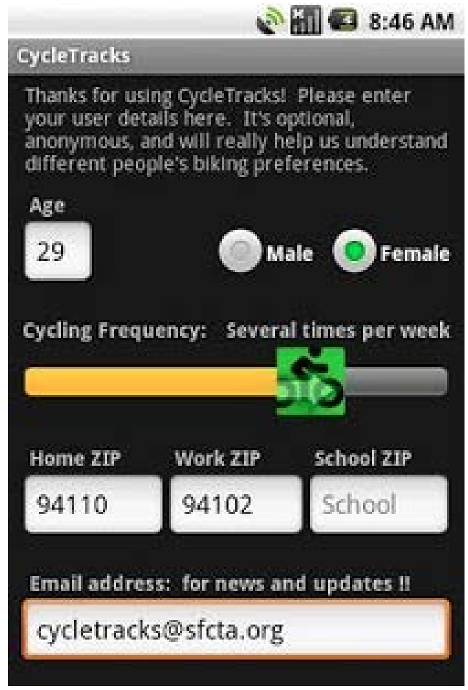
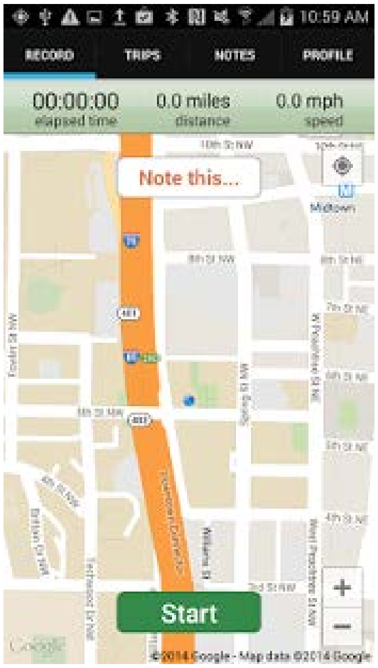


Figure 1.3: Cycle Tracks settings screen

Cycle Atlanta is based on the source code from Cycle Tracks but was updated for deployment in Atlanta. Cycle Atlanta was last updated in late 2014 and, therefore, has more modern user interface and several additional features, including a map view as shown in Figure 1.4, which enables users to see their bike routes after they are recorded. Additionally, it shows the history of the user's trips, as shown in Figure 1.5. Cycle Atlanta also includes more options for collecting demographics information in a profile screen, as shown in Figure 1.6. Users can also take notes about their trips.



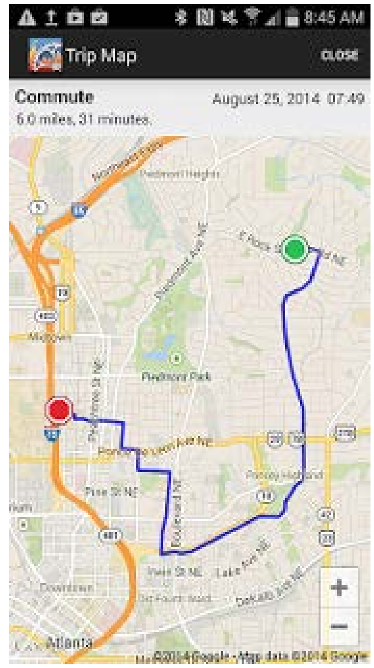


Figure 1.4: Cycle Atlanta main screen

Figure 1.5: Cycle Atlanta trip details

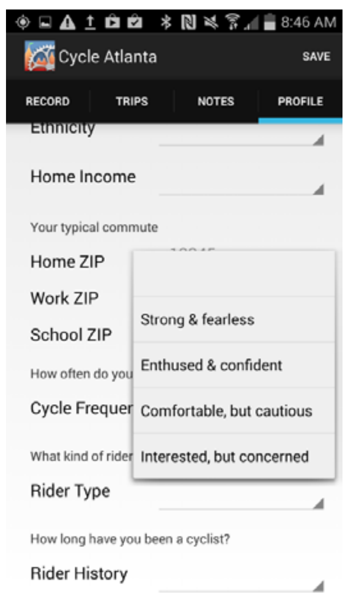


Figure 1.6: Cycle Atlanta profile settings

Table 1.1 shows the additional demographic and bicyclist profile information that can be collected in Cycle Atlanta, as compared to Cycle Tracks. The additional demographics allow researchers and practitioners to segment the data to determine if particular groups have different travel patterns, such as if certain rider types prefer route detours (e.g., "strong and fearless" prefers a direct route). The "Notes" feature was added to provide the City of Atlanta with another source of data on infrastructure issues. The assets were added to maps that Cycle Atlanta produces (e.g., to show bike racks or cut-throughs that cyclists use).

Cycle Atlanta offers several new features beyond Cycle Tracks and was updated more recently; therefore, the Cycle Atlanta source code was used as a basis for the multi-region application developed in this project.

Table 1.1: Collected information, Cycle Tracks and Cycle Atlanta

Cycle Tracks	Cycle Atlanta
● Demographics ○ Age ○ Gender ○ Cycling frequency ○ ZIP codes	● Demographics ○ Age ○ Gender ○ Cycling frequency ○ ZIP codes ○ Ethnicity ○ Home income ○ Rider type ○ Rider history ● Notes (issues & assets)

2.0 MULTI-REGION SOFTWARE ARCHITECTURE

Originally, the Cycle Atlanta system architecture was a single-server design that handled both the website and mobile phone apps for just one city, Atlanta. To expand the Cycle Atlanta mobile apps to more easily serve multiple regions, the architecture required updating.

To extend the existing Cycle Atlanta mobile apps to be available in new cities, they were modified to support multiple “regions”—using this design, Atlanta is one region and Tampa is another. This multi-region architecture allows multiple cities to use the same mobile apps downloaded from the Apple App Store and Google Play. This multi-region architecture concept is illustrated in Figure 2.1.

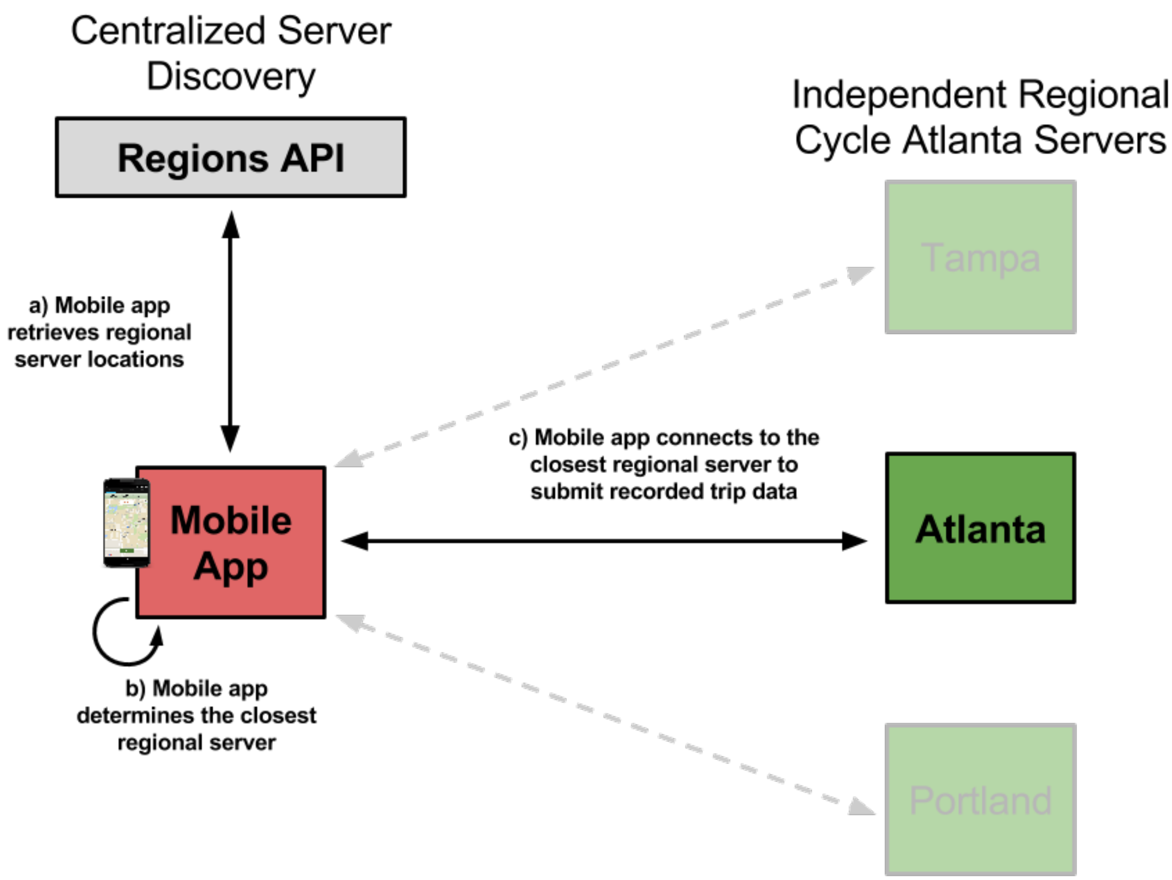


Figure 2.1: Multi-region architecture overview

To support multi-region apps, a centralized Bike Region Server Directory is used to hold a list of Cycle Atlanta servers, with one server per region. A Regions Application Programming Interface (API) makes this server information available to mobile apps.

Figure 2.2 shows the detailed communication protocol that takes place for a mobile device to discover and retrieve information from Regions API and the regional server. After the user opens the app, the app retrieves regional information from the Region API. Then, it will determine the current region of the app by checking the device's current location against regional bounds. When the closest server is selected, the app communicates with and submits data to the local regional server.

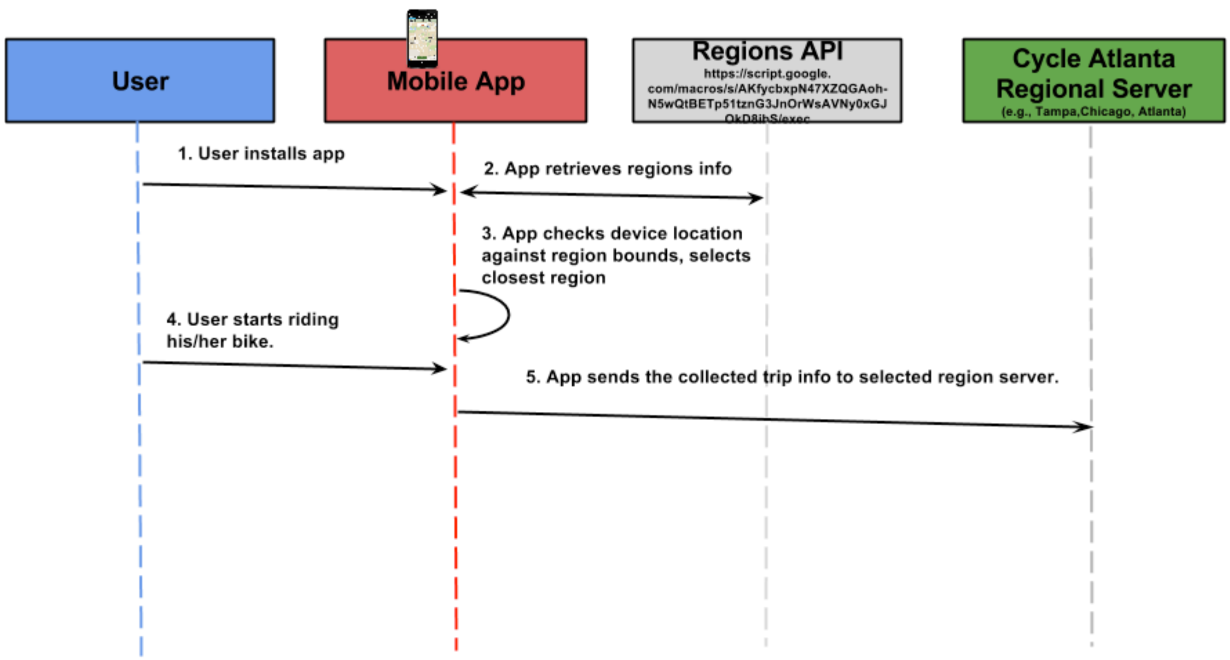


Figure 2.2: Communication protocol for app to discover and communicate with regional server

3.0 PROOF-OF-CONCEPT IMPLEMENTATION

3.1 MOBILE APPS

A multi-region architecture was implemented using modified versions of the Cycle Atlanta iOS and Android applications, an addition of 14,543 and 7,928 lines of code, respectively. Source code from the multi-region features of the OneBusAway iOS and Android features was used in this project, under the Apache v2.0 license.

On first startup, the mobile apps will retrieve a list of currently available regions and their coverage areas from the Regions API. Figure 3.1 shows the proof-of-concept implementation of the multi-region architecture in Cycle Atlanta Android and iOS mobile apps; the "Settings" screen shows a list of regions that are available.

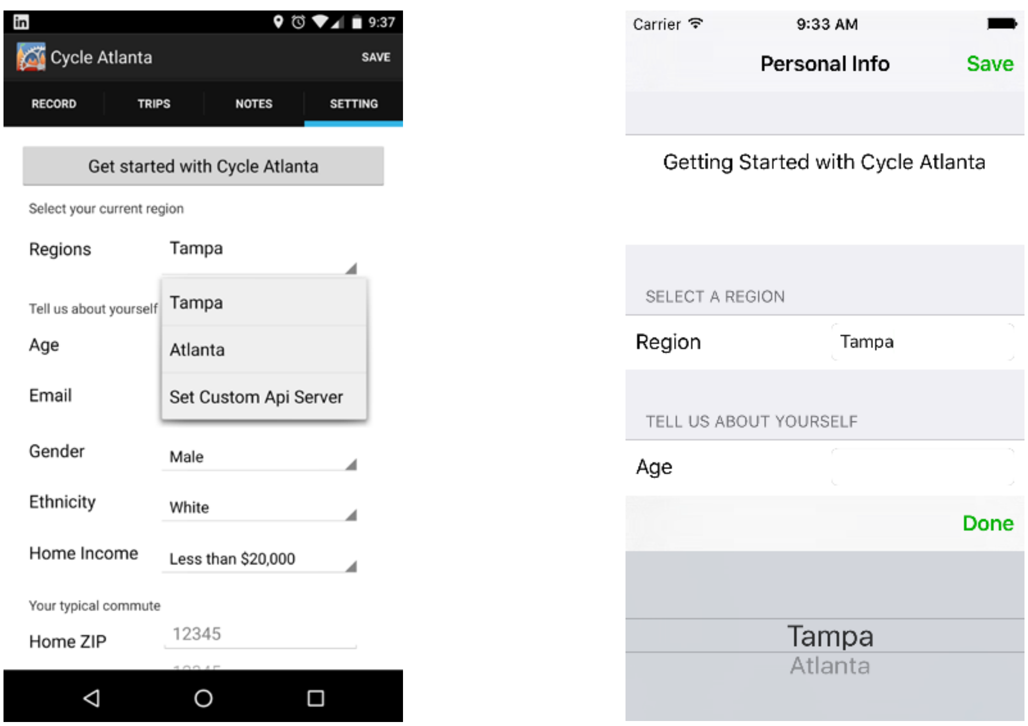


Figure 3.1: Multiple regions shown in multi-region Cycle Atlanta Android (*left*) and iOS (*right*) mobile apps

Once the list of currently available regions and their coverage areas have successfully been retrieved from the Regions API, the active servers will be compared to the user's real-time location (from either cellular, WiFi, or GPS positioning systems). If the user's real-time location falls within the bounds of an existing region, then the map view will focus on that region and the API of that region will automatically be used so that the multi-region feature of the app is completely transparent to the end user. If the user's real-time location is not available, or if the user is not within any existing coverage areas, the user will be presented with a list of available regions that provide all required APIs from which to choose.

3.2 REGIONS API

The Bike Regions Server Directory is implemented in a Google Sheet online (Figure 3.2), which offers an easy-to-use interface and access controls and permissions for editing the contents. The full Google Sheet can be viewed at https://docs.google.com/spreadsheets/d/1g9ROmJh-jhQxU_YfxeovIfAx9EAb3MEvpROx8Aa1-u4/.

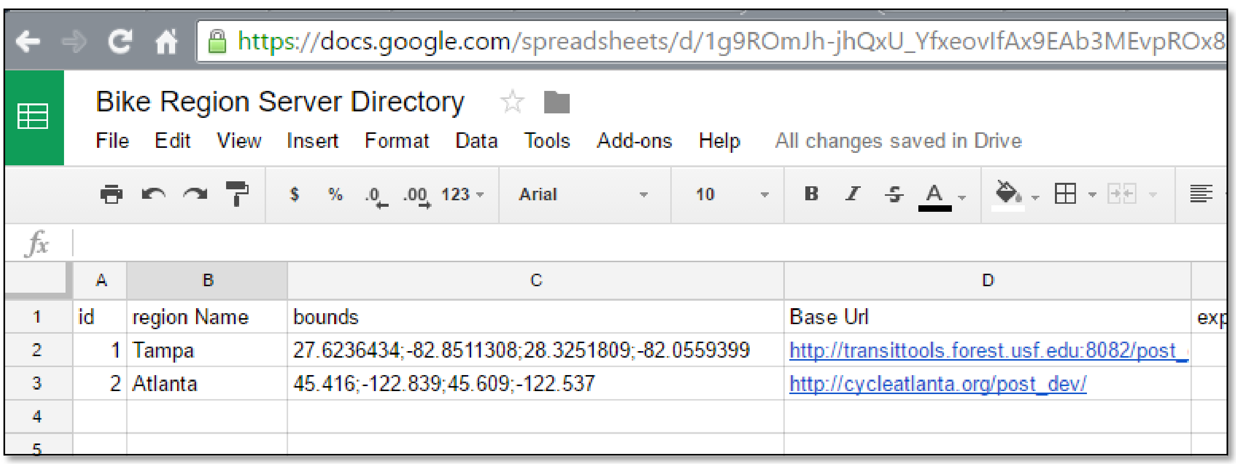


Figure 3.2: Bike Regions Server Directory

In this proof-of-concept implementation, two regions are shown—Tampa and Atlanta. Tampa features a Cycle Atlanta server that was set up as part of this project, and Atlanta uses the original Cycle Atlanta server deployed in Atlanta.

Table 3.1 shows the available server directory fields. Several fields (e.g., Open311 server URL for issue reporting) are shown as "future use." These fields are not currently used in the multi-region Cycle Atlanta mobile apps but have proven useful in the multi-region design of OneBusAway and, therefore, could be added to the Cycle Atlanta apps at a later date.

Table 3.1: Bike Region Server Directory fields

Server Directory Field	Explanation
Region Name	General name of region covered by Cycle Atlanta server; mobile app users can select this server; will be name of server shown
Base URL	Public base URL for entry point of directory and real-time APIs of regional server
Bounds	Geographic bounds of regional server
Active	TRUE if this server is active
Contact Email	Contact person's email address for region; mobile app users can send email inside app to maintainer of server being used to report issue or send feedback
Twitter URL (optional) (future use)	URL of Twitter feed for region
Facebook URL (optional) (future use)	URL of Facebook page for region
Experimental (future use)	TRUE if server is experimental and has not been approved by community as "production" server; FALSE if it appears as production region in mobile apps
Tutorial URL	URL of instructions for app for this region; accessible if user goes to "Settings" and taps on "Getting Started with the app" button
Open311 Base URLs (optional) (future use)	Optional URL where crowd-sourced issue reports can be submitted via Open311 format

Google Sheets also offers a method to automatically export the contents of the server directory spreadsheet to a format that is accessible to mobile apps. The contents of this directory are made available to mobile apps via a Google Apps Script (approximately 300 lines of code),[1] which was created to export the contents of the direction in the JavaScript Object Notation (JSON) format. This JSON output is referred to as the Regions Application Programming Interface (API). The contents of this Regions API[2] is shown in Figure 3.3.

[1] https://script.google.com/d/1WyTLiZcG3JkkHMzjsqhyTpyxCQJ5YrgScUe4JynjbjOFJwoHcLfgtJxL/edit?usp=sharing
[2] https://script.google.com/macros/s/AKfycbxpN47XZQGAoh-N5wQtBETp51tznG3JnOrWsAVNy0xGJOkD8ibS/exec

Figure 3.3: Regions API accessed by multi-region mobile apps

3.3 ADDING A NEW REGION

It is expected that additional regions will be added to the multi-region architecture deployment. Therefore, a process to add and edit regions is needed.

Adding new regions is a two-step process:

1. The region sets up their own Cycle Atlanta server. Source code and instructions for deploying a local Cycle Atlanta server can be found at https://github.com/CUTR-at-USF/cycleatlanta.org.

2. A multi-region administrator adds the new region to the server directory, which then becomes available in the Regions API.

The implementation of the server directory using Google Sheets and the Google Apps Script makes the second step very simple; the administrator can add a new row to the spreadsheet with the new information for a new region or edit an existing row. These changes are automatically exported to the Region API by the Google Apps Script as soon as the edit or a new row is completed. This process is shown in Figure 3.4.

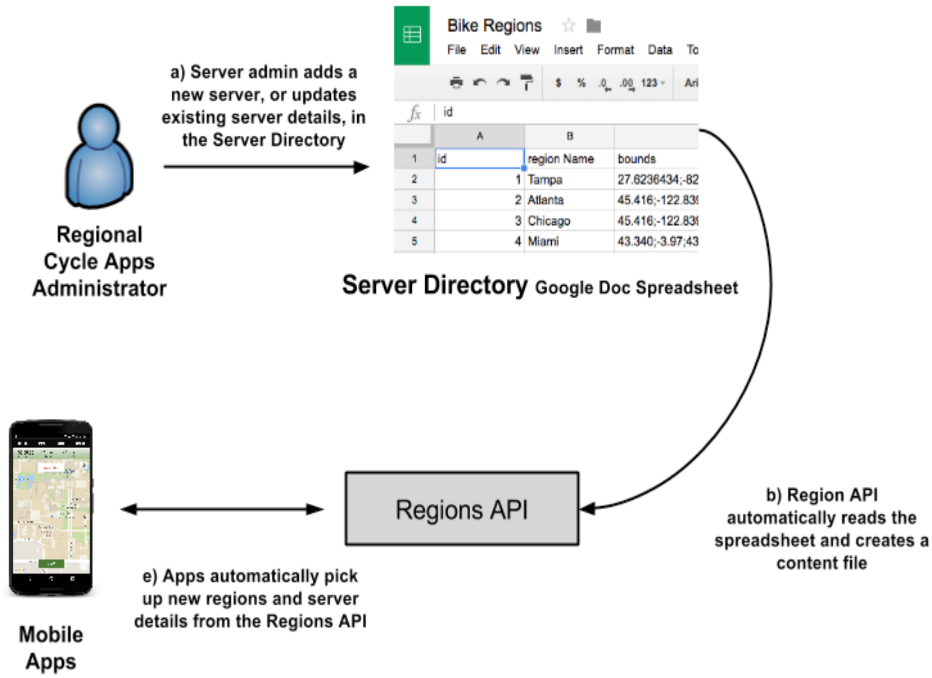


Figure 3.4: Multi-region administration via Google Sheets

3.4 SAMPLE DATA COLLECTION

The Cycle Atlanta server software was deployed to a server at USF in Tampa to test the multi-region design. Both the original Atlanta server and the Tampa server were then added to the server directory. The iOS and Android apps were then started, and a trip was recorded with each, with live locations in Tampa and simulated locations in Atlanta. In both cases, the apps were able to identify the nearest server correctly and submit recorded trip data. The Tampa website with two sample recorded trips is shown in Figure 3.5 (trips are orange paths), and Figure 3.6 shows the same two trips within the Cycle Atlanta multi-region Android app.

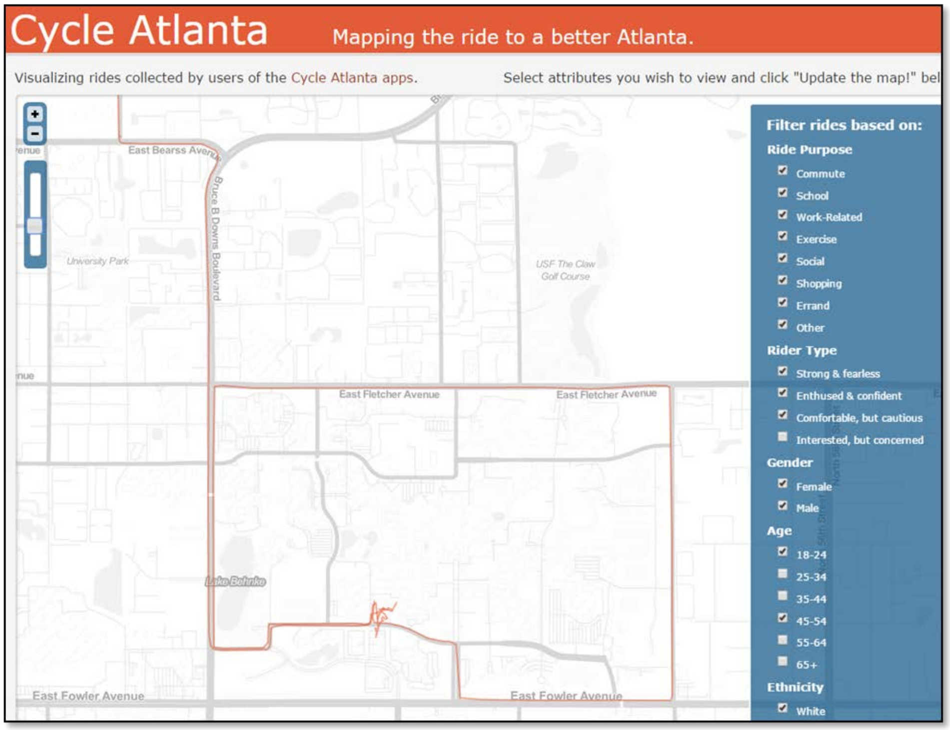


Figure 3.5: Sample bike data recorded in Tampa using multi-region app, shown on the Tampa website

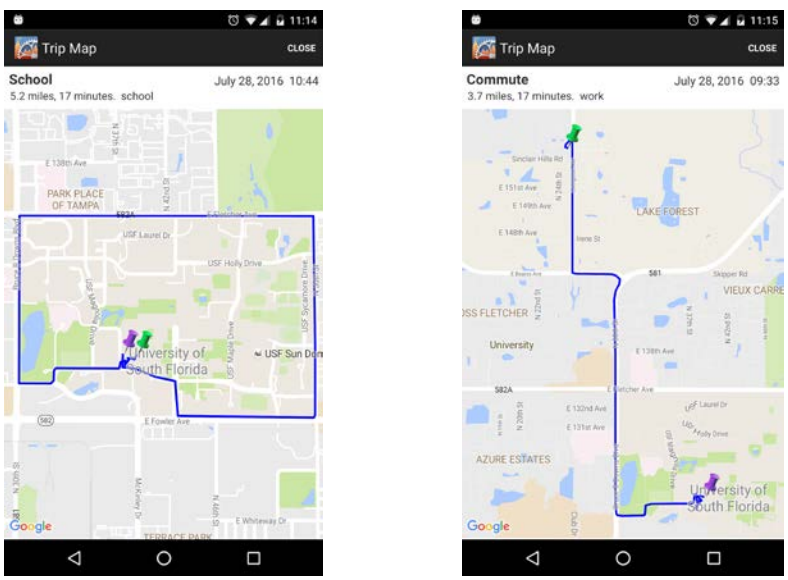


Figure 3.6: Sample bike data recorded in Tampa using multi-region app, shown on the mobile device

4.0 CONCLUSIONS AND FUTURE WORK

4.1 CONCLUSIONS

A proof-of-concept, multi-region architecture for the Cycle Atlanta Android and iOS apps was designed and implemented in this project. The source code for the multi-region versions of the Cycle Atlanta Android and iOS apps is available online via the Center for Urban Transportation Research (CUTR) at the University of South Florida (USF) GitHub account, as shown in Figures 4.1 and 4.2. Github is a popular online repository for open-source projects, and also facilitates the sharing of code between related projects. The URLs to access the code are https://github.com/CUTR-at-USF/Cycle-Atlanta-Android and https://github.com/CUTR-at-USF/Cycle-Atlanta-Android, respectively, which also include additional information for configuring, compiling, and executing the applications. The compiled version of the most recent multi-region Cycle Atlanta Android app is available to download from https://github.com/CUTR-at-USF/Cycle-Atlanta-Android/blob/regions/apk/Cycle%20Atlanta.apk.

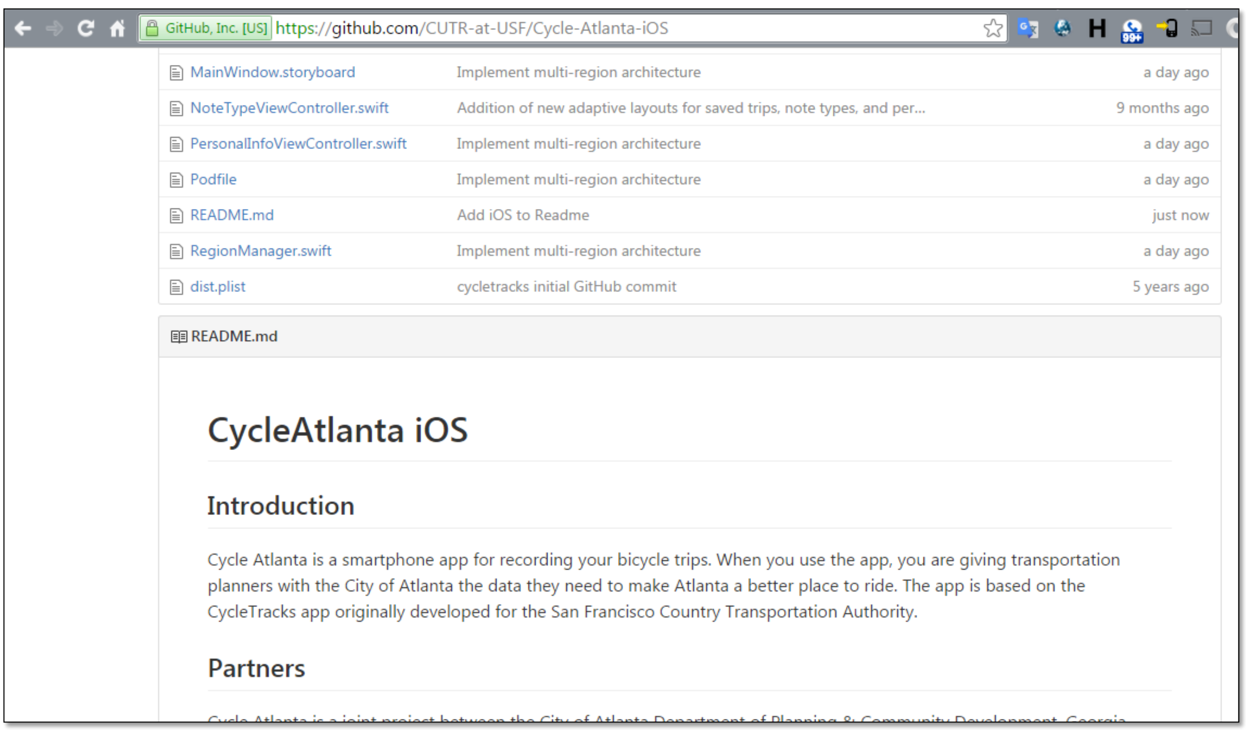


Figure 4.1: Multi-region Cycle Atlanta iOS GitHub page

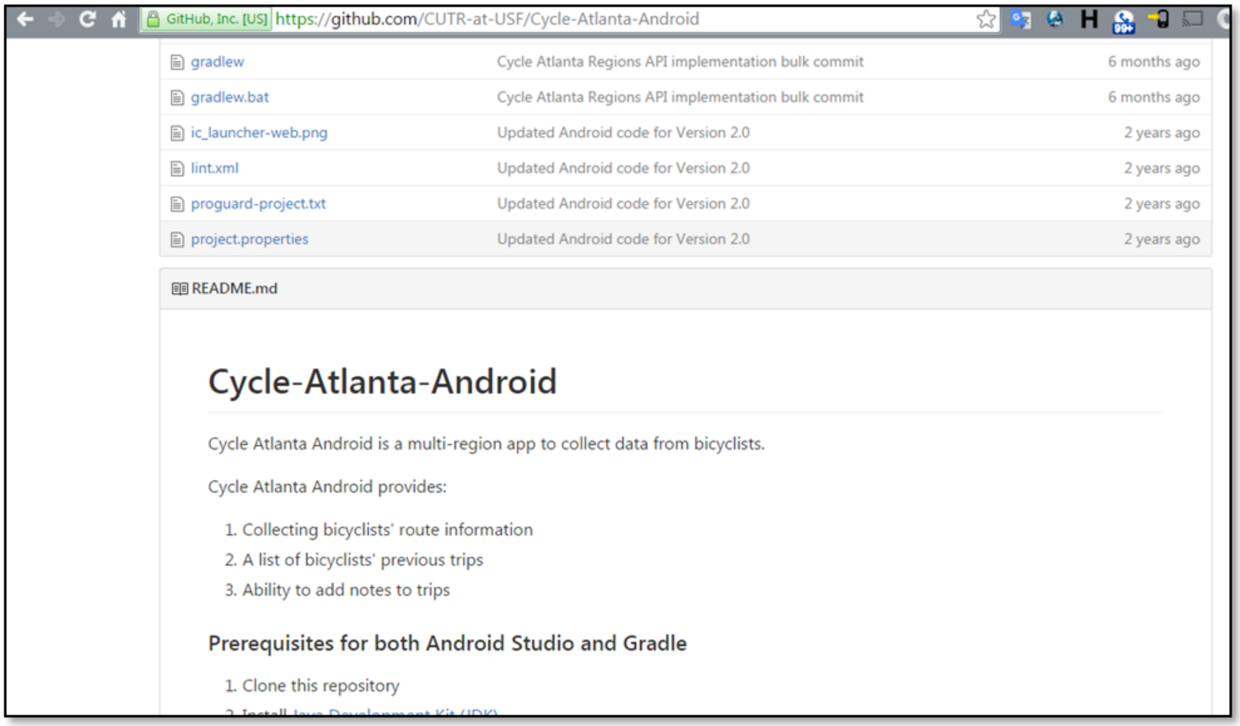


Figure 4.2: Multi-region Cycle Atlanta Android GitHub page

4.2 FUTURE WORK

This project represents the initial step towards a multi-region deployment of the Cycle Atlanta apps. Future work can expand on this proof-of-concept project with full deployments of the crowd-sourcing bike data apps in multiple cities, including the collection of real data from bicyclists. As part of this next step, a brand identity will need to be created that reflects the multi-region nature of the apps. While the software in this project was based on the source code for the Cycle Atlanta apps, the name “Cycle Atlanta” obviously implies a limited area of geographic coverage. It is recommended that the research team consult with SFCTA, Georgia Tech, and other potential stakeholders for the multi-region apps to determine what this brand identity could be. It may be possible to leverage the existing “Cycle Tracks” brand, although this would require direct coordination with SFCTA to leverage their existing infrastructure (e.g., server, Google Play and Apple App Store accounts). The SFCTA server would also need to be updated to use the Georgia Tech Cycle Atlanta version in order to support the data collection of new fields added by Georgia Tech.

Future work can also demonstrate the centralized data collection from the multi-region apps in a single location (e.g., the current Cycle Tracks database maintained by SFCTA, the NITC database for bike trip count data). SFCTA is currently allowing other regions to use the original Cycle Tracks app to record data, with this data being submitted back to SFCTA’s database. This model has the advantage of being able to analyze data across geographic regions, and does not require the new jurisdiction to deploy its own apps or server. However, this model requires direct coordination with SFCTA (including possible cost-sharing agreements among multiple regions to maintain the server as data and traffic grow in size), and also requires the local jurisdiction to be comfortable with an external entity (SFCTA) holding data from users in the local jurisdiction's area. In contrast, other regions (such as Atlanta) have chosen to create copies of the Cycle Tracks apps and maintain its own database. This decentralized model has the advantage of requiring no coordination with SFCTA, and allows the local region to have full control and ownership of all data collected from local users. However, the disadvantage of this model is that the local jurisdiction must maintain a local server. The multi-region architecture created in this project can support both centralized and decentralized models of data collection, as shown in Figure 4.3.

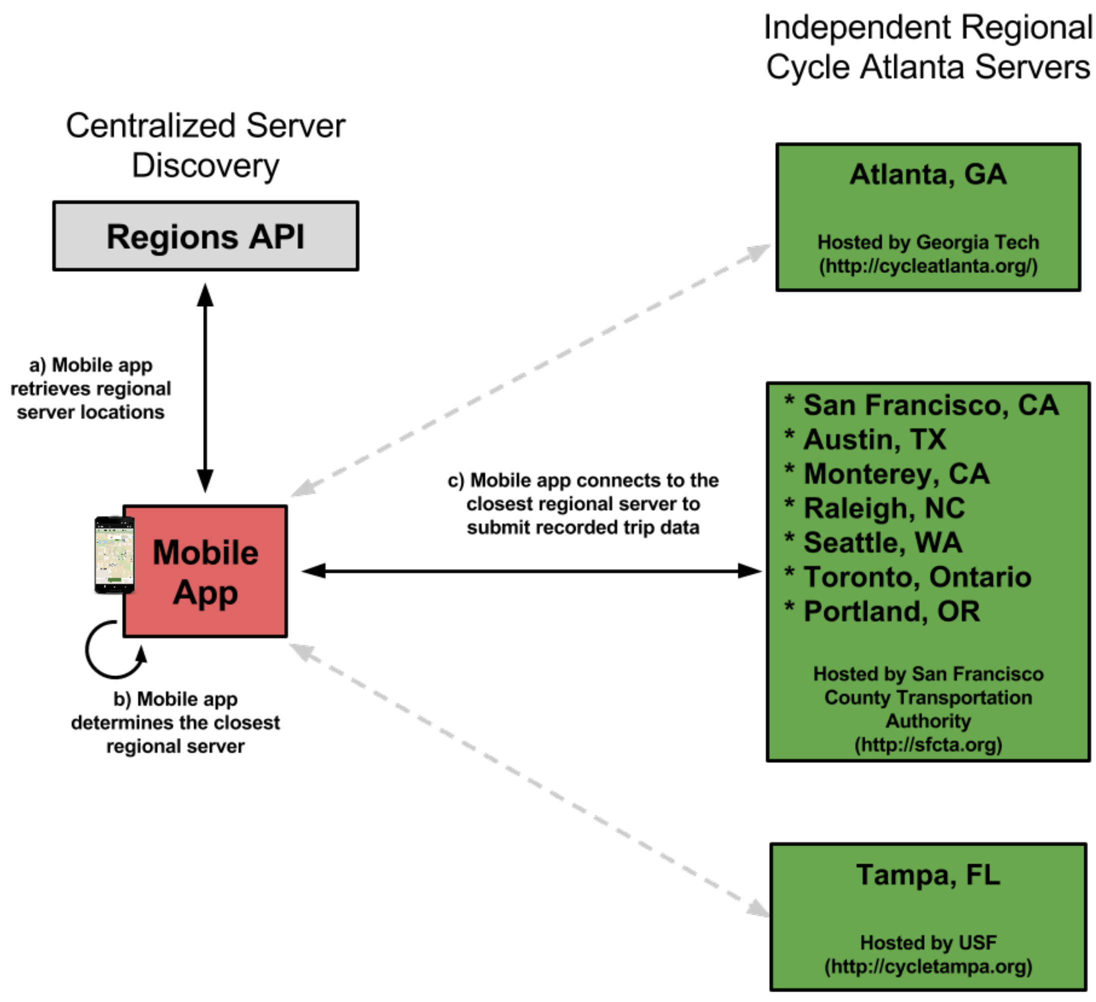


Figure 4.3: Possible deployment plan to support multiple regions' data in same server

To store data for more than one region in the same server, a new record would be created in the server directory and the Regions API with the geographic bounds for the new region. However, the server URL for that new region would be the same as another existing region. For example, in Figure 4.3 Atlanta and Tampa have each chosen to collect data in their own servers. However, San Francisco, Austin, Monterey, Raleigh, Seattle, Toronto, and Portland have all decided to share the same server, which is hosted by SFCTA. When the mobile apps start up and retrieve the Regions API data, if a user is in one of those regions corresponding with the SFCTA server, the app submits the data to SFCTA server; if the user is in the Atlanta region, the app submits the data to Cycle Atlanta server; and if the user is in the Tampa region, the app submits data to the USF server.

Compatibility with an older version of the apps is another area for future work. Currently, the multi-region architecture has been developed based on the Cycle Atlanta source code; however, the Cycle Atlanta mobile app collects more information than the Cycle Tracks mobile application, as discussed in Section 1.0. In order for SFCTA and other regions using older versions of the software to participate in the multi-region architecture and collect the new data fields created for Cycle Atlanta, these regions would need to update their servers to use the Cycle Atlanta server software. These regions would then gain the capability of collecting the new demographics and assets/notes fields added for Cycle Atlanta, and would also benefit from new improvements to the multi-region version of the apps going forward.

The Open311 standard (Open311, 2016) should also be investigated as a possible format for submitting assets and notes data collected by the Cycle Atlanta multi-region apps to issue management systems used by cities such as Washington, D.C., Boston, Chicago, and many others (Open311, 2016). There are a number of vendor-based and open-source solutions for dedicated issue management tools that support the Open311 protocol. If the Cycle Atlanta multi-region apps supported the submission of issues via Open311, it would allow cities to leverage existing issue management systems to be able to triage and resolve problems reported by the public. A similar project investigating the integration of Open311 support with the OneBusAway multi-region apps to capture transit-specific issues is currently being conducted by CUTR at USF with funding from the National Center for Transit Research and the Florida Department of Transportation (National Center for Transit Research, 2016).

Finally, future work could also examine integrating the multi-region bike data recording functionality into other applications that provide ongoing value to the user. Users of the Cycle Tracks and Cycle Atlanta apps primarily record data for the purpose of civic duty and improving bike infrastructure within a city. However, the manual recording of such data can be burdensome to users, as it requires them to interact with the app each time they ride somewhere. The same data recording functionality could be integrated into other apps, such as OneBusAway, that provide users ongoing value. In the case of OneBusAway, this ongoing value is real-time transit arrival information. If additional bike-related functionality is added to OneBusAway to encourage the use by bicyclists (e.g., real-time bike-share information), bike trip data could be recorded via that app, potentially without requiring ongoing user interaction. While passive data recording is much less burdensome to the user, it also doesn't directly collect more detailed information such as trip purpose or mode of transportation (assuming multimodal travel). Past techniques have examined extracting trip purpose and mode of transportation using automated data mining techniques (Barbeau, 2010; Gonzalez, 2010), which could also be used here. A less intrusive minimalist survey technique, where the user is intelligently prompted for small pieces of information at opportune times (e.g., asking the user how they got to a bus stop when they are waiting for a bus), could also be examined.

5.0 REFERENCES

Barbeau, S. J., A. Borning, and K. Watkins. “OneBusAway Multi-region: Rapidly Expanding Mobile Transit Apps to New Cities.” *Journal of Public Transportation,* 17(4), 14-34. 2014.

Barbeau, S.J., N. Georggi, P.L. Winters. “TRAC-IT: Travel Behavior Data Mining using GPS-enabled Mobile Phones,” U.S Department of Transportation Federal Highway Administration Travel Model Improvement Program (TMIP) National Webinar on Data Transferability and Data Mining, January 25, 2010.

Cycle Atlanta. http://CycleAtlanta.org. 2015.

Gonzalez, P.,J. Weinstein, S. Barbeau, M. Labrador, P.L. Winters, N. Georggi, R. Perez. “Automating Mode Detection for Travel Behavior Analysis by Using GPS-enabled Mobile Phones and Neural Networks,” Institution of Engineering and Technology (IET) Intelligent Transportation Systems, 2010, Vol. 4, Iss. 1, pp. 37–49. doi: 10.1049/iet-its.2009.0029.

National Center for Transit Research. “Improving Access to Transit through Crowdsourced Information,” http://www.nctr.usf.edu/2015/03/improving-access-to-transit-through-crowdsourced-information/. 2016.

Open311. http://www.open311.org/. 2016.

Portland State University. “ORcycle.” http://www.pdx.edu/transportation-lab/orcycle. 2015.

Ride Report. “Ride Report – Count Every Ride, Automatically,” https://ride.report/. 2016.

San Francisco County Transportation Authority. “Moving the City: Cycle Tracks for iPhone and Android.” http://www.sfcta.org/modeling-and-travel-forecasting/Cycle Tracks-iphone-and-android. 2015.

Watkins, K., R. Ammanamanchi, J. LaMondia, C. Le Dantec. “Comparison of Smartphone-based Cyclist GPS Data Sources,” Proceedings of 2016 Transportation Research Board Conference, Washington, D.C., January 9-13, 2016.

NITC
NATIONAL INSTITUTE for TRANSPORTATION and COMMUNITIES
Transportation Research and Education Center
Portland State University
1900 S.W. Fourth Ave., Suite 175
Portland, OR 97201