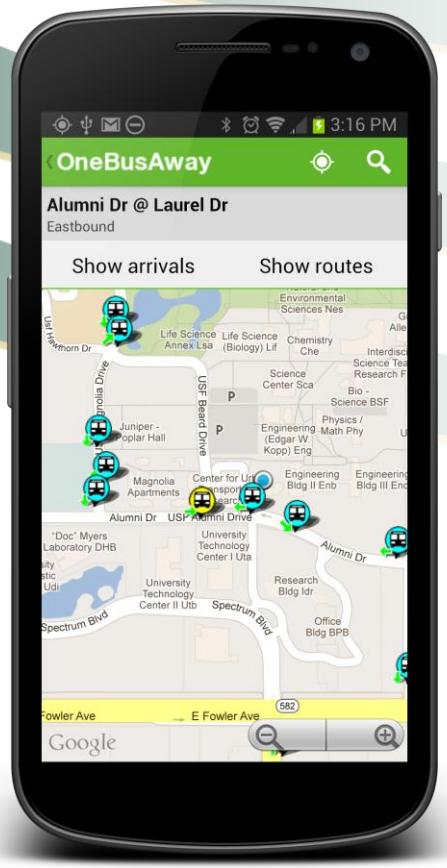


CUTR

CENTER for URBAN TRANSPORTATION RESEARCH

Open Transit Data – A Developer's Perspective

Sean J. Barbeau, Ph.D.

Overview

- Why Open Data?
- Anatomy of Transit Data Sharing
- Being Developer-Friendly

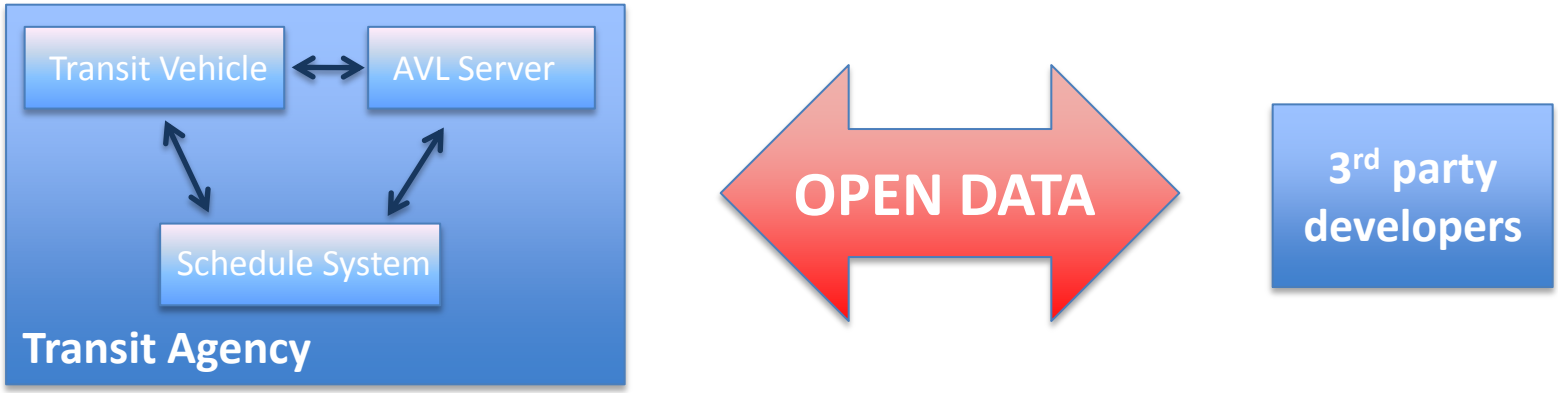
WHY OPEN DATA?

What is open data?

- Transit data that is shared with the public
  - Typically shared via website/FTP site/web services
    - No login should be required (may use API key)
  - Should be updated **regularly**, with any changes in schedule/routes/stops

Open [Data ≠ Architecture ≠ Source]

- ***Open architectures*** mostly focus on:
  - Standards **within** an agency's software/hardware systems
  - Interconnectivity with other government systems
- ***Open source*** means software source code is available
- ***Open data*** is the sharing of data with **external public parties**

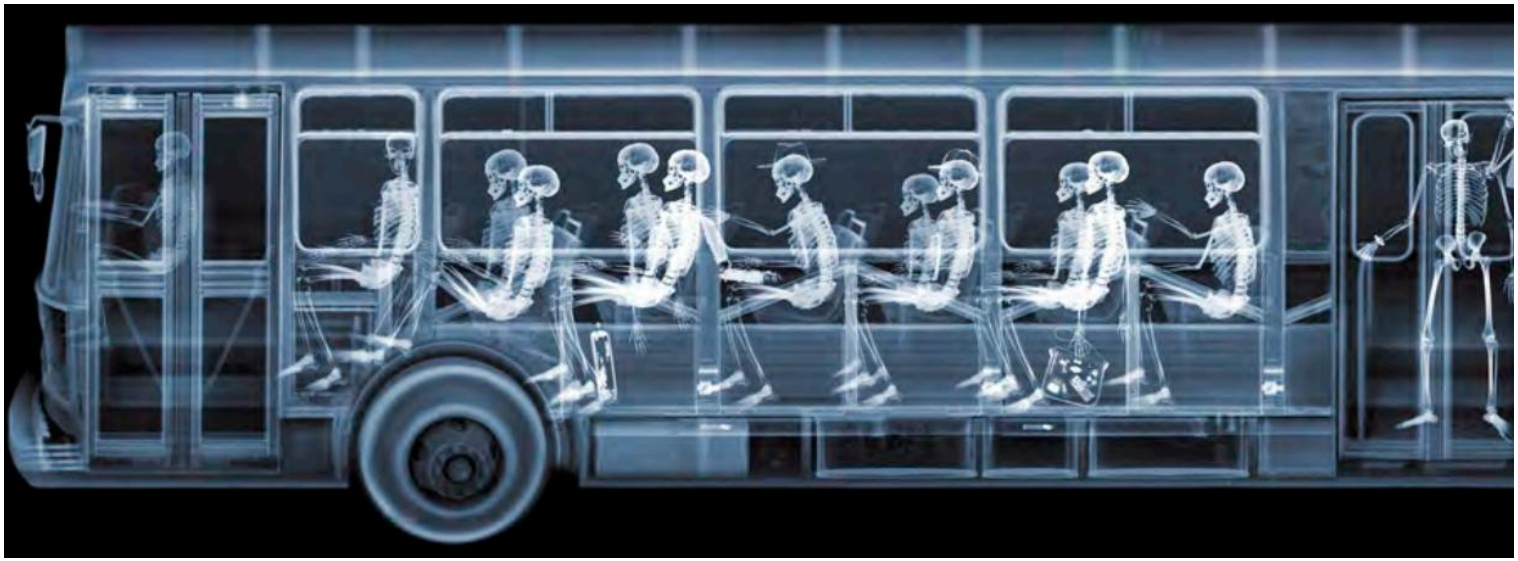


Why is open data important?

- Allows public to contribute services that are cost/time-prohibitive for the public sector
  - e.g., many mobile platforms
- Vendors are unpredictable
  - Some agencies have shared data only with Google
  - When Apple dropped Google Maps, iPhone users lost transit directions
  - Apple relied on 3rd party apps to fill the gap – only possible if open data was available

Why is open data important (to developers)?

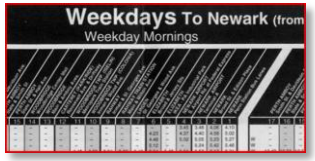
- Developers want to create innovative apps that meet a need!
  - Some are monetized, some are not
- If you don't provide open data, developers will often improvise
  - …via website scraping, etc.
  - Prone to breaking
  - Not beneficial to agency or rider



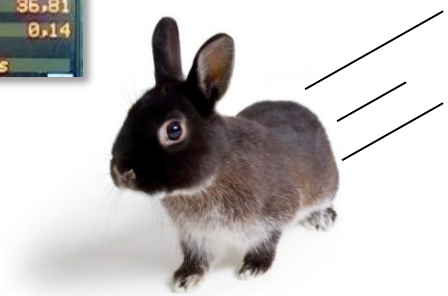
THE ANATOMY OF TRANSIT DATA SHARING

Two Types of Open Data

1. **Static**
   - e.g., Transit schedules / routes / stops
   - Change only a few times a year



2. **Real-time**
   - e.g., Estimated arrival times /vehicle positions/service alerts
   - Can change every few seconds



Two Magnitudes of Open Data

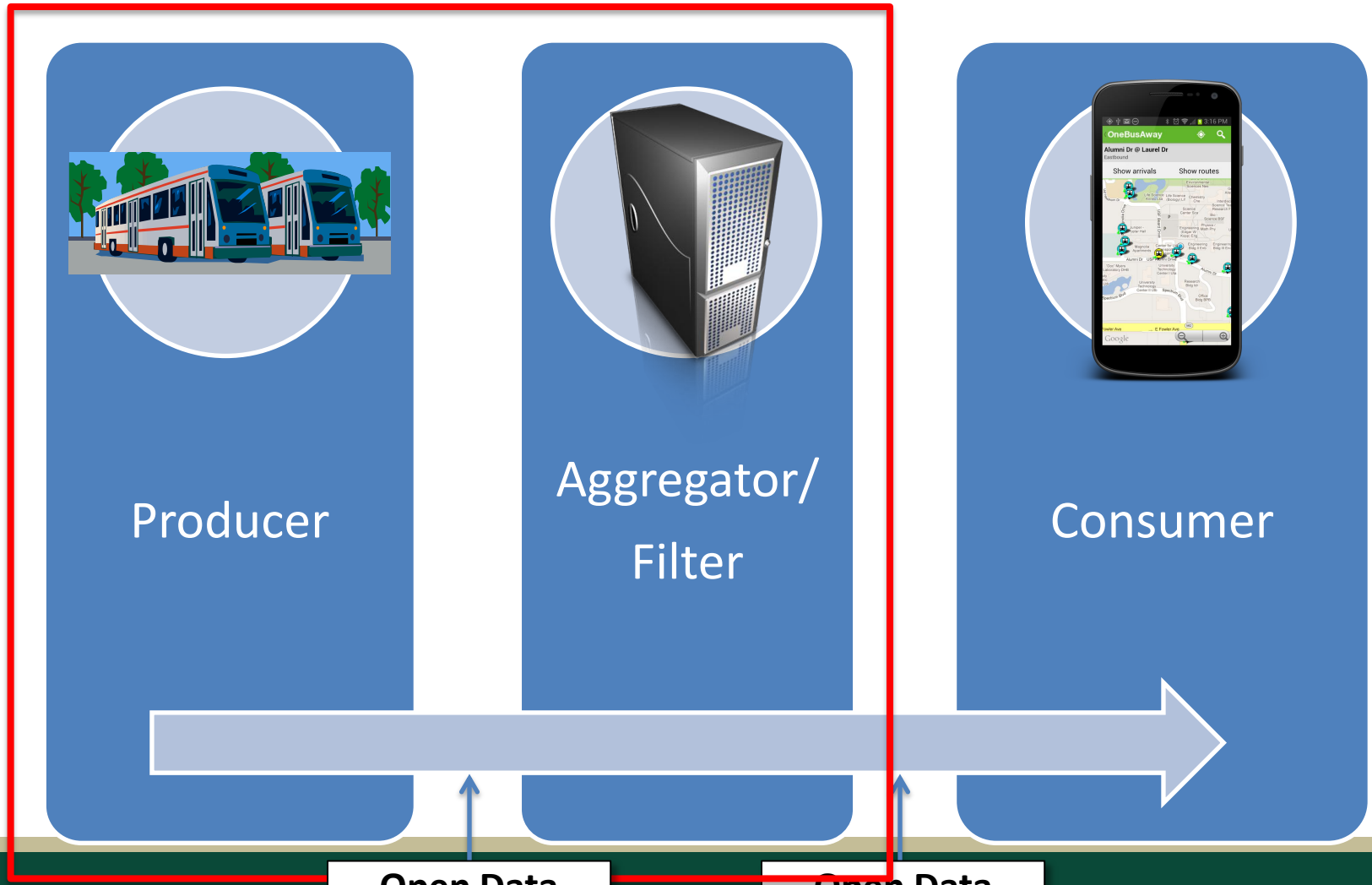
A. "Fire hose"

- A dump of the complete state of the transit system
- **Not directly** suitable for mobile devices
  - **Static** -> All transit schedules/routes/stops
  - **Real-time** -> All estimated arrivals/vehicle positions/service alerts

B. "Faucet"

- Precise subset of transit data
- Suitable for mobile devices
  - **Static ->** "Stop ID 10 is served by Route 5"
  - **Real-time ->** "It is 2 minutes until Route 5 bus arrives at Stop ID 10"

Transit Data Flow Architecture



Commonly-used "fire hose" formats

- static
- realtime

Producer

Transit Communications Interface Profiles (TCIP)

General Transit Feed Spec. (GTFS)

GTFS-realtime

Service Interface for Real time Information (SIRI)

Aggregator/ Filter

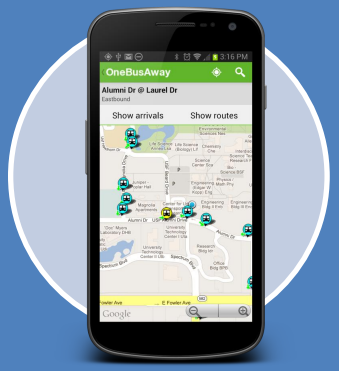
Open Data ("Firehose")

GTFS/GTFS-realtime format - http://goo.gl/tmwv8
SIRI format – http://goo.gl/Vnpyv
TCIP format - http://goo.gl/vd6kY

Transit Data Flow Architecture

Producer

Aggregator/
Filter



Consumer

Open Data
("Fire hose")

Open Data
("Faucet")

Common "faucet" formats still emerging

- static
- realtime

Aggregator/
Filter

Vendor/Agency-specific formats

OneBusAway API

SIRI
(REST/JSON format)

Vendor/Agency-specific formats

OneBusAway API

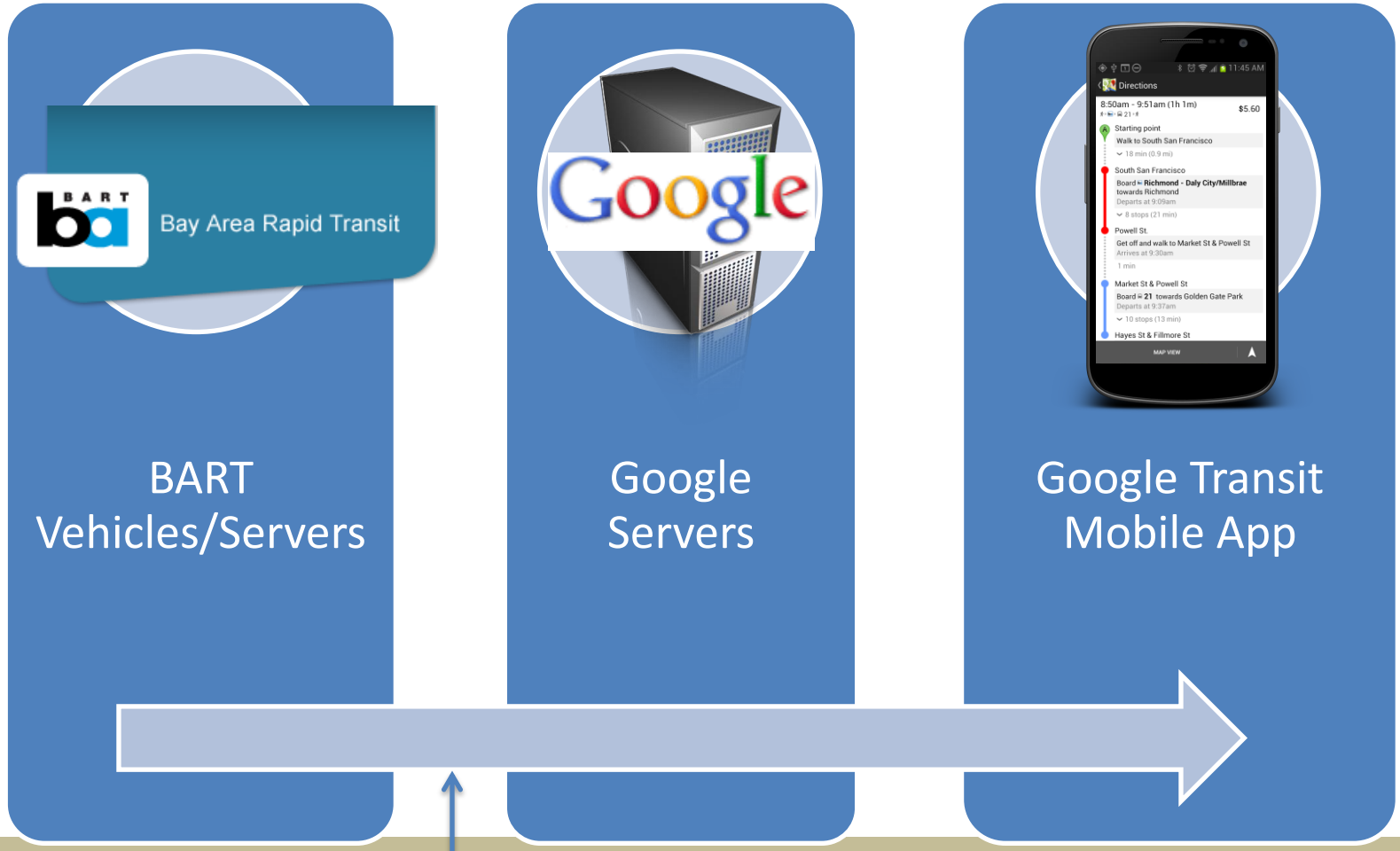
Consumer

Open Data ("faucet")

Vendor/Agency formats - http://goo.gl/NtNJ0
OneBusAway format - http://goo.gl/XXJyN
SIRI REST format - http://goo.gl/0PctT

Example – ~~Google Transit~~

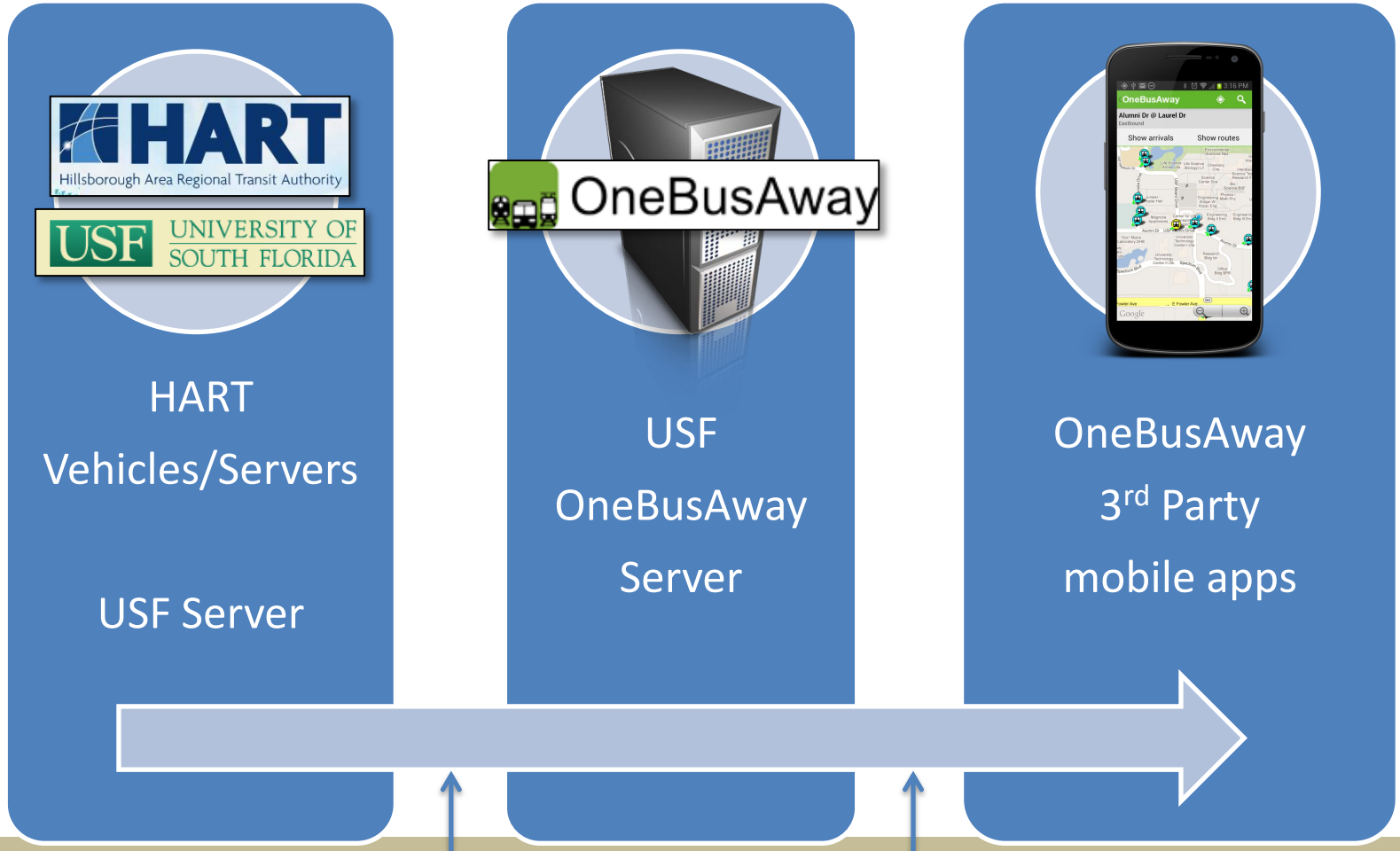
Any 3rd party



Static - GTFS
Realtime - GTFS-realtime

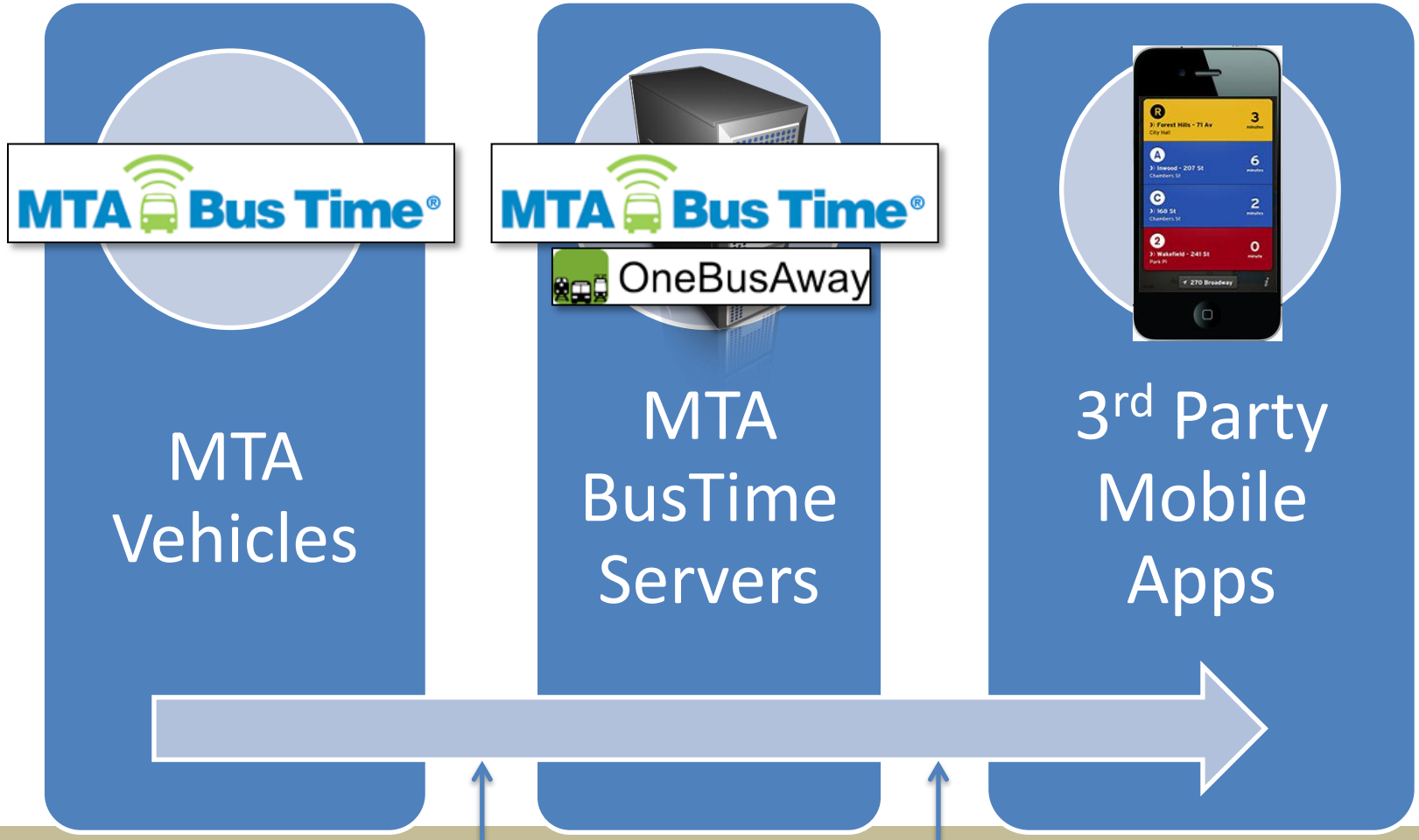
Open to Public

Example – HART in Tampa, FL



More at http://goo.gl/iqHD2

Example – MTA BusTime in NY

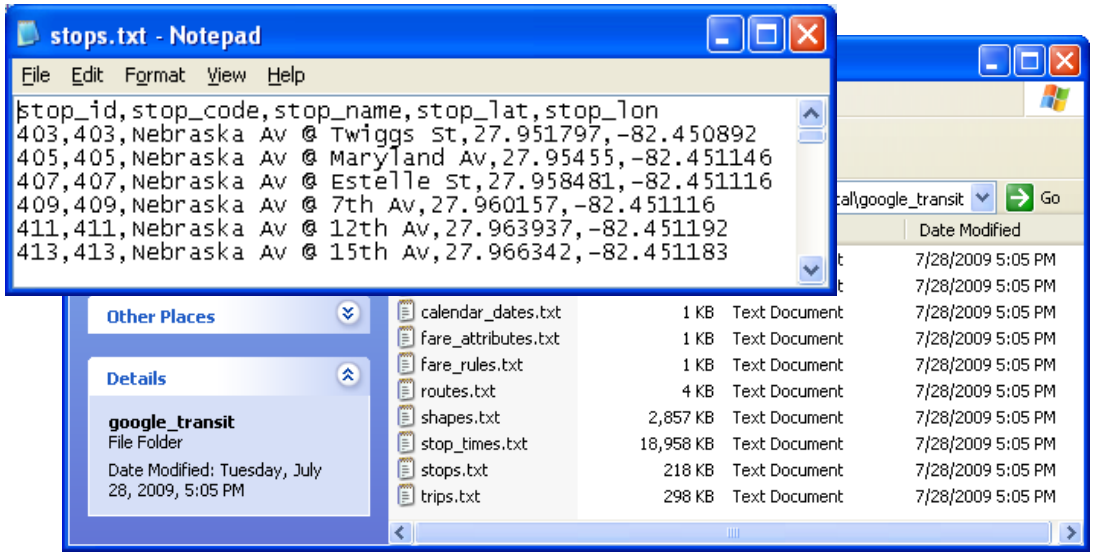


Successful Open Data Formats Are…

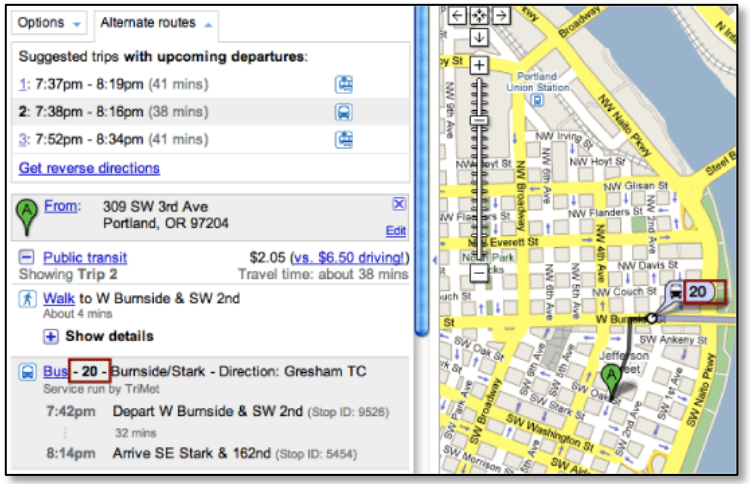
- **Organic**
  - Created and improved by the people actually producing and consuming the data
- **Open**
  - Open process for evolution
  - Data/documentation not hidden behind log-ins
- **Easy-to-use** for **app developers**
  - Is documentation simple to understand?
  - Are there existing open-source software tools?

General Transit Feed Specification (GTFS)

- Created by TriMet and Google in 2005
- Has become a *de facto* standard world-wide for static transit schedule/route/stop data



GTFS data consists of multiple text files



GTFS data powers Google Transit and other apps

General Transit Feed Specification (GTFS)

- Over 500 agencies worldwide have transit data in GTFS format[1]
  - 49 of top 50 largest U.S. transit agencies share GTFS data, over 227 worldwide
  - At least 20 Canadian agencies share open data
- Most agencies created GTFS data for Google Transit
  - But, GTFS is open-data format used by web/mobile apps, OpenTripPlanner, OneBusAway, etc.[2]
- See "GTFS Data Exchange" for list of agencies with GTFS data
  - http://www.gtfs-data-exchange.com/
  - Or, ask your local agency

[1] City-Go-Round, http://www.citygoround.org/, Dec. 4, 2012
[2] For more GTFS info and references, see paper co-authored by Sean Barbeau and Aaron Antrim – "The Many Uses of GTFS Data" - http://goo.gl/asR96

Promoting app development with open data

BEING DEVELOPER-FRIENDLY

Create a relationship with developers

- Open your GTFS data, and share on GTFS-Data-Exchange!
  - GTFS data should not be password or login protected
- Share real-time data too (national list pending)
- Create a “Developer page” with access to resources (e.g., GTFS license, data)
- Create developer email list/group for announcements/Q&A/collaboration
- Announce resources on “Transit Developers” group[1]

HART Developer page - http://www.gohart.org/developers/

[1] Transit Developers group, https://groups.google.com/forum/?fromgroups#!forum/transit-developers, Dec. 4th, 2012

Be Developer-Friendly!

- Use a simple “Terms of Service” based on existing industry examples[1][2][3][4][5]
- Use GTFS naming conventions throughout
  - “Direction_ID” is 0/1 (**not** N/S/E/W) in real-time data too!
- Make sure IDs match among datasets
  - E.g., tripID in real-time data matches GTFS tripID

[1] TriMet “Terms of Use." http://developer.trimet.org/terms_of_use.shtml
[2] BART "Terms of Use." http://www.bart.gov/dev/schedules/license.htm
[3] Corona, CA "Terms of Use.” http://www.discovercorona.com/City-Departments/Public-Works/Transportation/GTFS.aspx
[4] PSTA "Terms of Use.” http://www.psta.net/developers/License%20Agreement%20for%20App%20Devs.pdf
[5] HART "Terms of Use.” http://www.gohart.org/developers/terms_of_use.html

Be Developer-Friendly!

- Use developer/mobile-friendly formats
  1. For data – GTFS, GTFS-realtime, SIRI REST API (see MTA NY BusTime API[1])
  2. For mobile APIs – RESTful web services design and JSON encoding preferred (**not** SOAP and XML)

SOAP Request

```
POST /busstoparrival/busstopws.asmx HTTP/1.1
Host: 73.205.128.123
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/GetNextNVehicleArrivals"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNextNVehicleArrivals xmlns="http://tempuri.org/">
    <n>int</n>
    <RouteID>int</RouteID>
    <DirectionCodeID>int</DirectionCodeID>
    <BusStopID>int</BusStopID>
     <TripID_External>string</TripID_External>
    </GetNextNVehicleArrivals>
  </soap:Body>
</soap:Envelope>
```

HTTP-Post Request

```
GET /busstoparrival/busstopws.asmx/ GetNextNVehicleArrivals?
n=string&RouteID=string&DirectionCodeID=string
&BusStopID=string&
TripID_External=string HTTP/1.1 Host: 73.205.128.123
```

- 3.7 times more characters using SOAP!

XML Response

```
<Siri xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns:ns4="http://datex2.eu/schema/1_0/1_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt"
xmlns="http://www.siri.org.uk/siri"> <ServiceDelivery>
<ResponseTimestamp>2012-09-12T09:28:17.213-
04:00</ResponseTimestamp> <VehicleMonitoringDelivery>
<VehicleActivity> <MonitoredVehicleJourney> <LineRef>MTA
NYCT_S40</LineRef> <DirectionRef>0</DirectionRef>
<FramedVehicleJourneyRef> <DataFrameRef>2012-09-
12</DataFrameRef> <DatedVehicleJourneyRef>MTA
NYCT_20120902EE_054000_S40_0031_MISC_437</DatedVehicleJou
rneyRef> </FramedVehicleJourneyRef> <JourneyPatternRef>MTA
NYCT_S400031</JourneyPatternRef>
<PublishedLineName>S40</PublishedLineName>
<OperatorRef>MTA NYCT</OperatorRef> <OriginRef>MTA
NYCT_200001</OriginRef> </MonitoredVehicleJourney>
</VehicleActivity> </VehicleMonitoringDelivery> <ServiceDelivery>
</Siri>
```

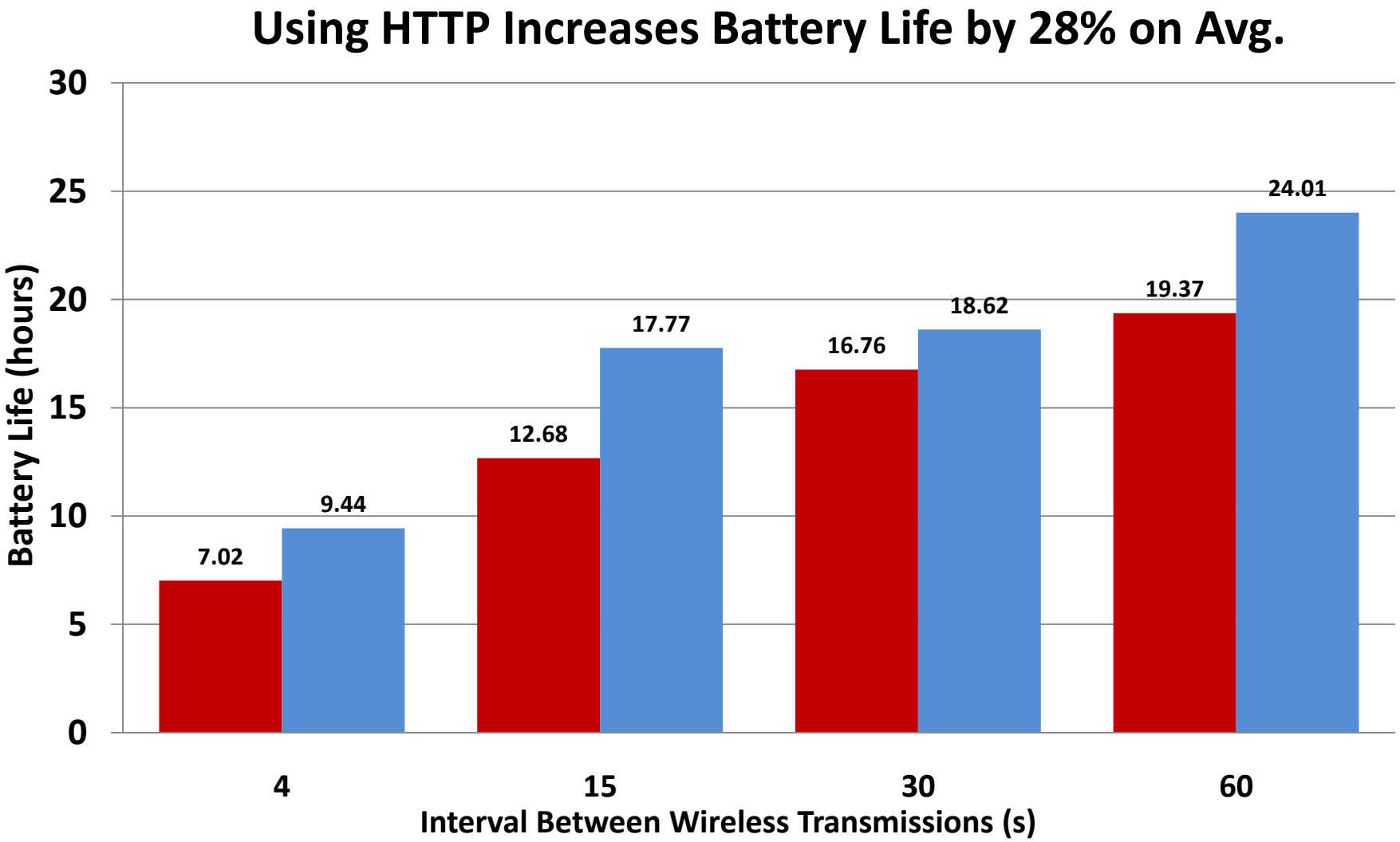
JSON Response

```
{ Siri: { ServiceDelivery: { ResponseTimestamp: "2012-08-
21T12:06:21.485-04:00", VehicleMonitoringDelivery: [ {
VehicleActivity: [ { MonitoredVehicleJourney: { LineRef: "MTA
NYCT_S40", DirectionRef: "0", FramedVehicleJourneyRef: {
DataFrameRef: "2012-08-21", DatedVehicleJourneyRef: "MTA
NYCT_20120701CC_072000_S40_0031_S4090_302" },
JourneyPatternRef: "MTA NYCT_S400031", PublishedLineName:
"S40", OperatorRef: "MTA NYCT", OriginRef: "MTA
NYCT_200001" } } ] } } } }
```

- 1.8 times more characters using XML!

[1] MTA BusTime API, http://bustime.mta.info/wiki/Developers/SIRIIntro, Dec. 4th, 2012

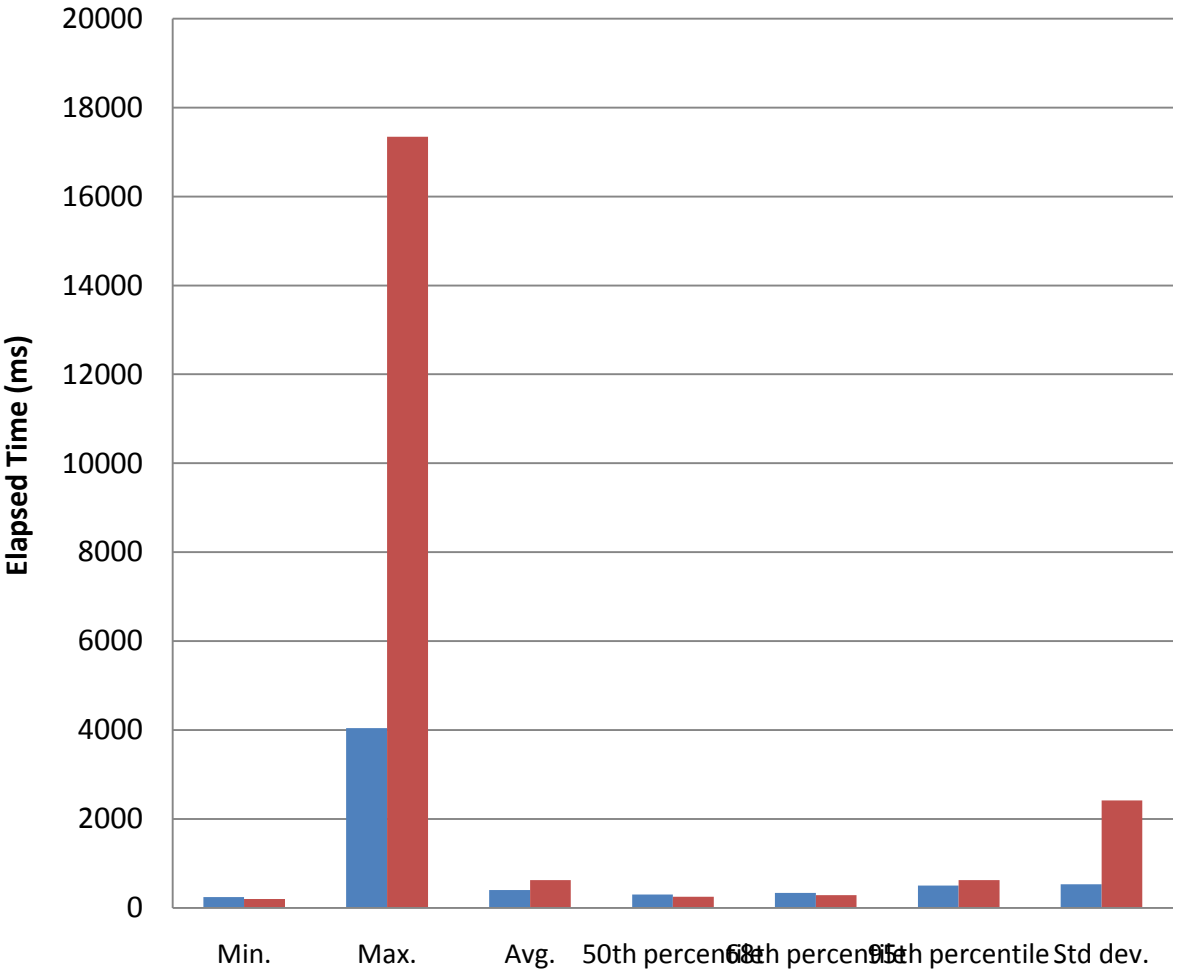
SOAP vs. HTTP



Motorola i580 phone

-See http://goo.gl/hq6nE for details

XML vs. JSON Parsing Time – Samsung Galaxy S3



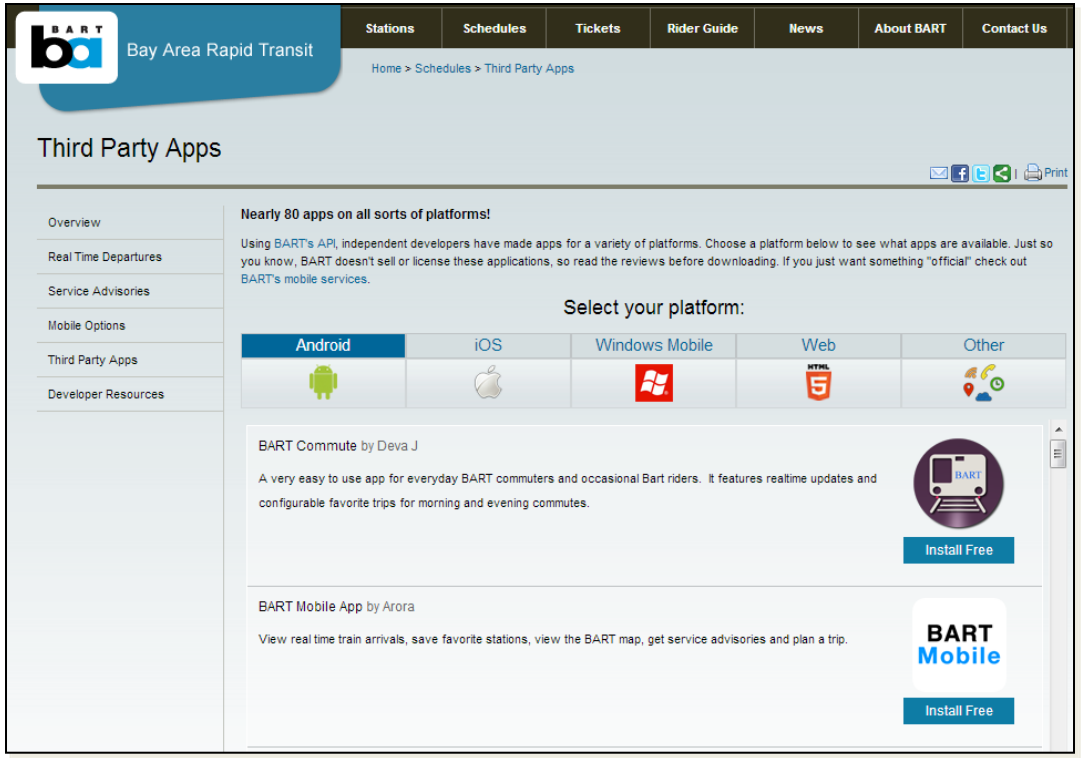
- **~4.3 times longer** to parse the first response using XML
- **First response time is critical for mobile apps**, since application state is often destroyed when user multitasks (checks email, etc.) on their phone

-Using Jackon 2.1.2
-Using MTA SIRI REST API StopMonitoring

-See http://goo.gl/EhYSl for details

Get the word out!

- After developers have created mobile apps, share them with riders
- Consider an “App Center”[1-9] to showcase apps



[1] TriMet "TriMet App Center." http://trimet.org/apps/
[2] BART "Third Party Apps." http://www.bart.gov/schedules/appcenter/
[3] MTA "App Center." http://www.mta.info/apps/
[4] CTA "App Center." http://www.transitchicago.com/apps/
[5] GoTriangle. "App Center." http://www.gotriangle.org/developers/transit_apps
[6] HART "App Center." http://www.gohart.org/developers/appcenter.html
[7] MBTA "App Center." http://www.mbta.com/rider_tools/apps/
[8] KCATA "App Center." http://www.kcata.org/maps_schedules/app_center/
[9] UTA"App Center." http://developer.rideuta.com/DeveloperApps.aspx

Conclusions

- Open data (e.g., GTFS) makes transit apps possible
- Understand open [data vs. architecture vs. source]
- Understand the differences in data:
  - Static vs. real-time
  - “Fire hose” vs. “Faucet”
- Understand that certain formats are more appropriate than others for certain situations (e.g., mobile)
- Being developer-friendly encourages mobile app development!

Thanks!

Sean J. Barbeau, Ph.D.

barbeau@cutr.usf.edu

813.974.7208

Principal Mobile Software Architect for R&D

Center for Urban Transportation Research

University of South Florida

For more GTFS info and references, see paper co-authored by Sean Barbeau and Aaron Antrim – "The Many Uses of GTFS Data" - http://goo.gl/asR96

CUTR

Glossary

- API – Application Programming Interface
- AVL – Automatic Vehicle Location
- FTP – File Transfer Protocol
- GTFS – General Transit Feed Specification
- HTTP – HyperText Transfer Protocol
- IT – Information Technology
- JSON – Javascript Object Notation
- REST – Representational State Transfer
- SIRI -Service Interface for Real time Information
- TCIP - Transit Communications Interface Profiles
- XML – Extensible Markup Language

CUTR