

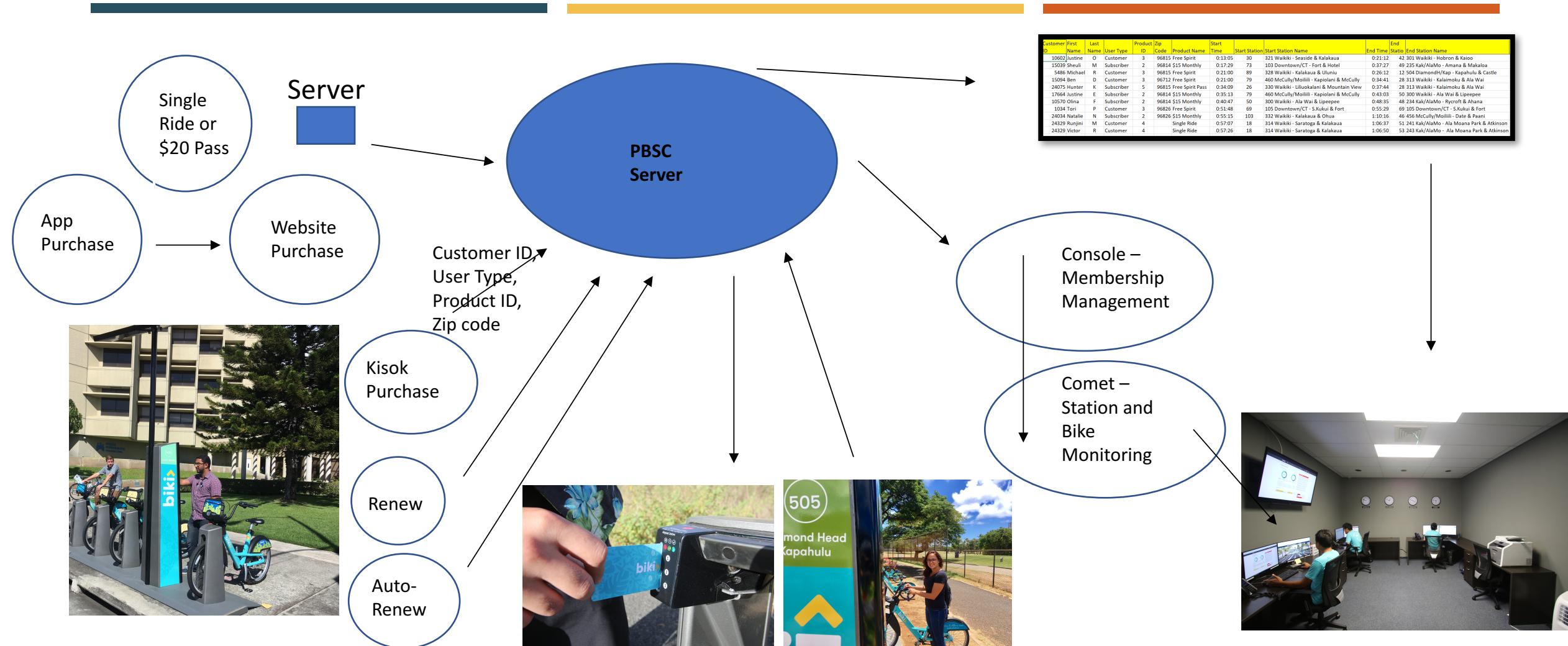
---

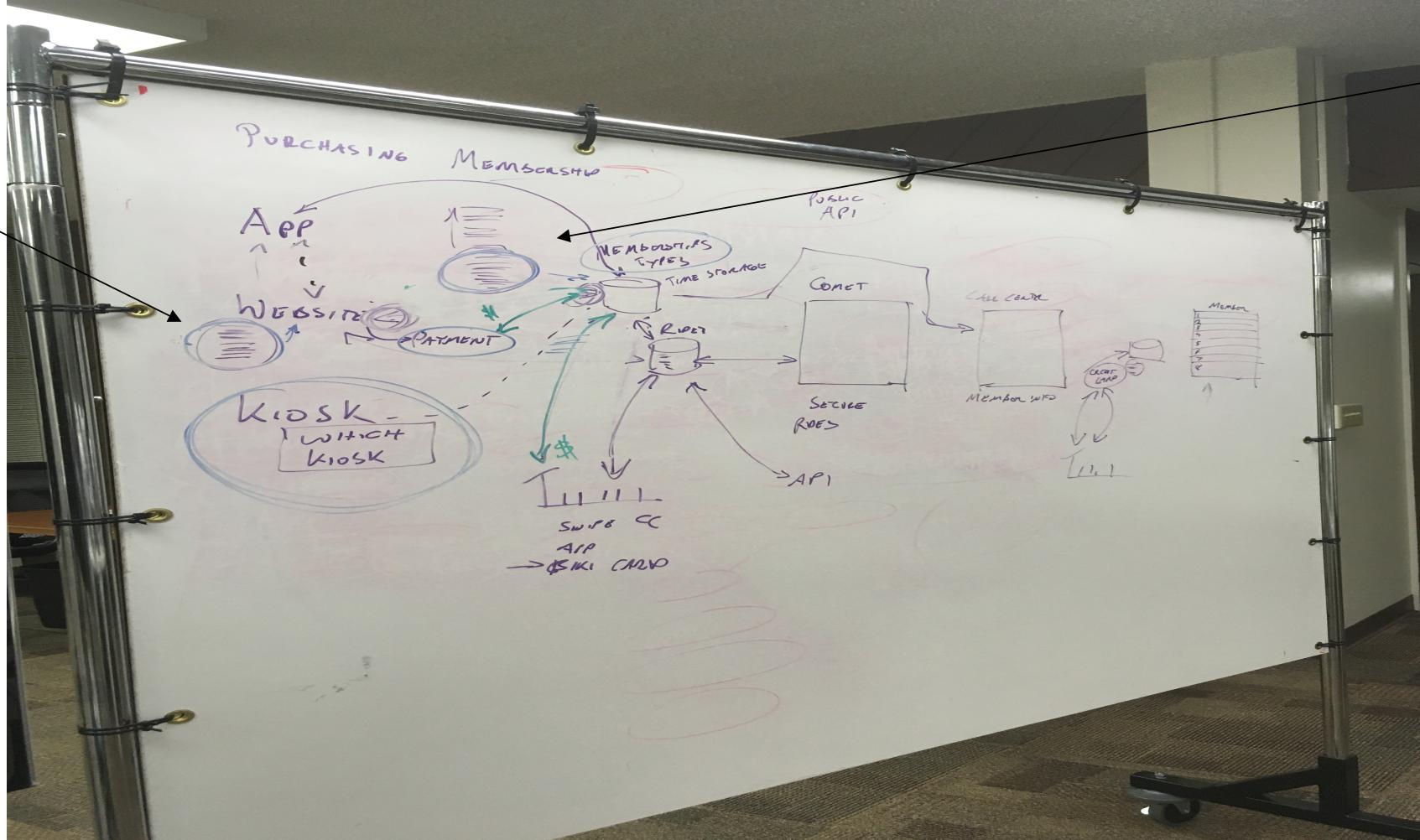
Sprint Review

# Biki Data Flow



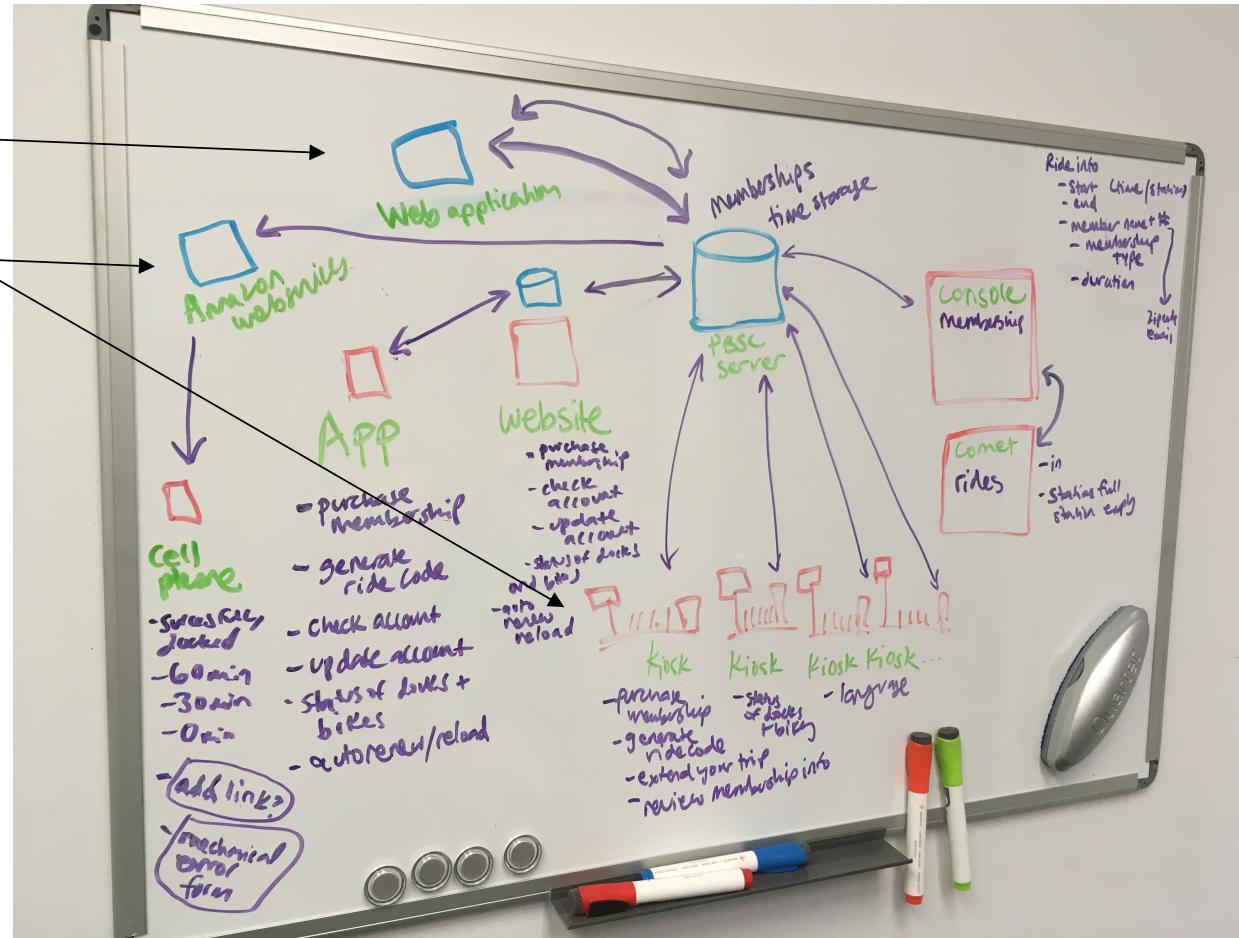
devleague



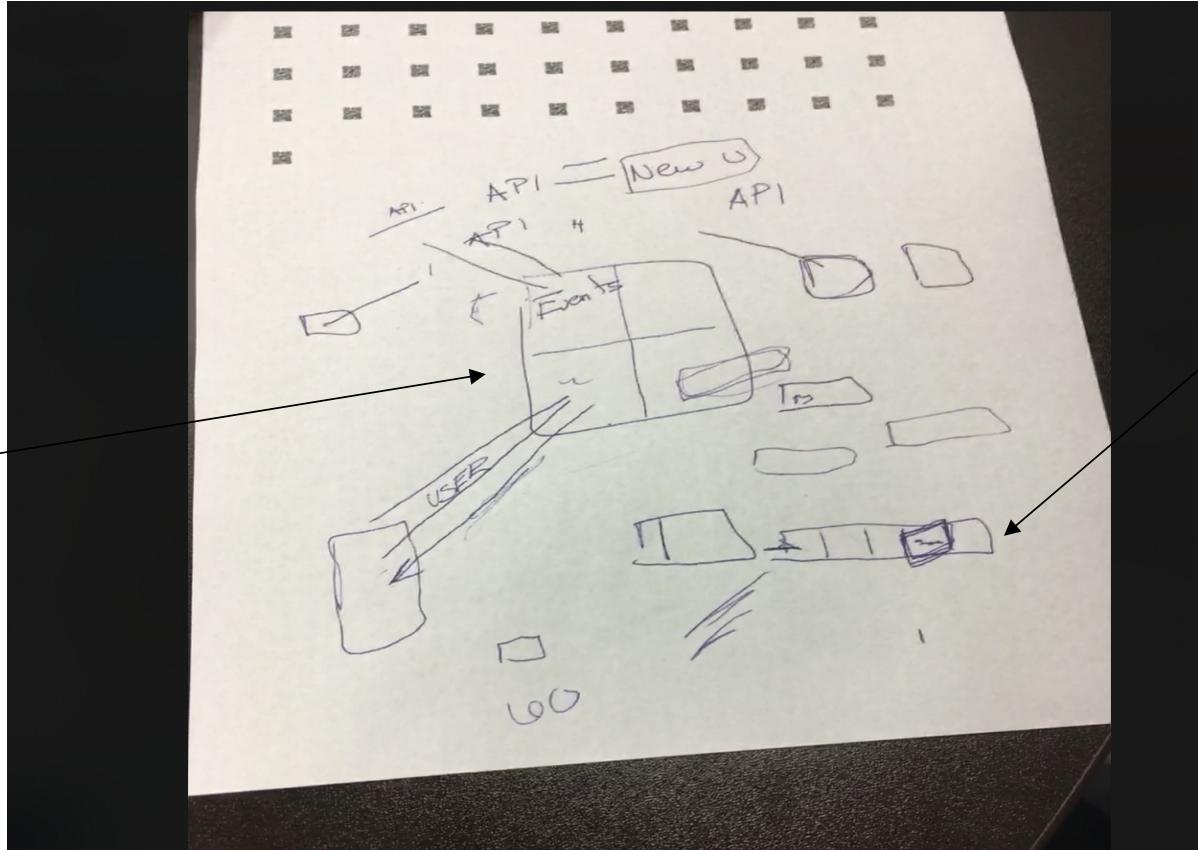


Here's where I first learned there exists a trail of data for everything. Just need to identify things clearly and ask the right questions

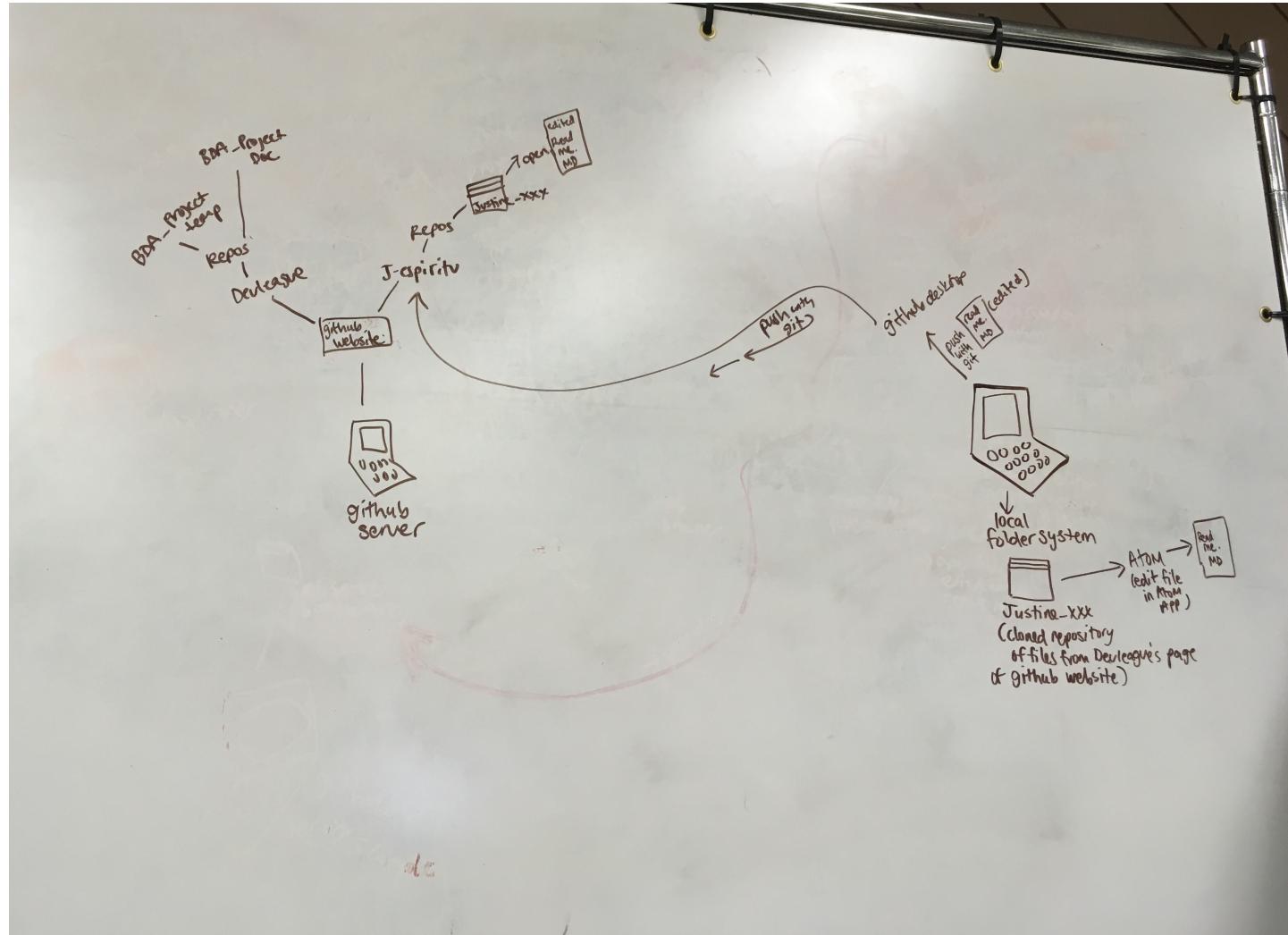
Every time I talk to someone else, seems like there's a new party that's holding the data I want



Our tech guy explained PBSC Server as a file cabinet holding folders where every folder is an event, and everything that happens is an event (a transaction, a screen getting touched or a button, or a customer service call). Some events have a "key" or an **API**. Our events about starting and ending trips has a public API. Events I want to track, PBSC would need to assign an API to them, and that's what transmits or gives access to the data?

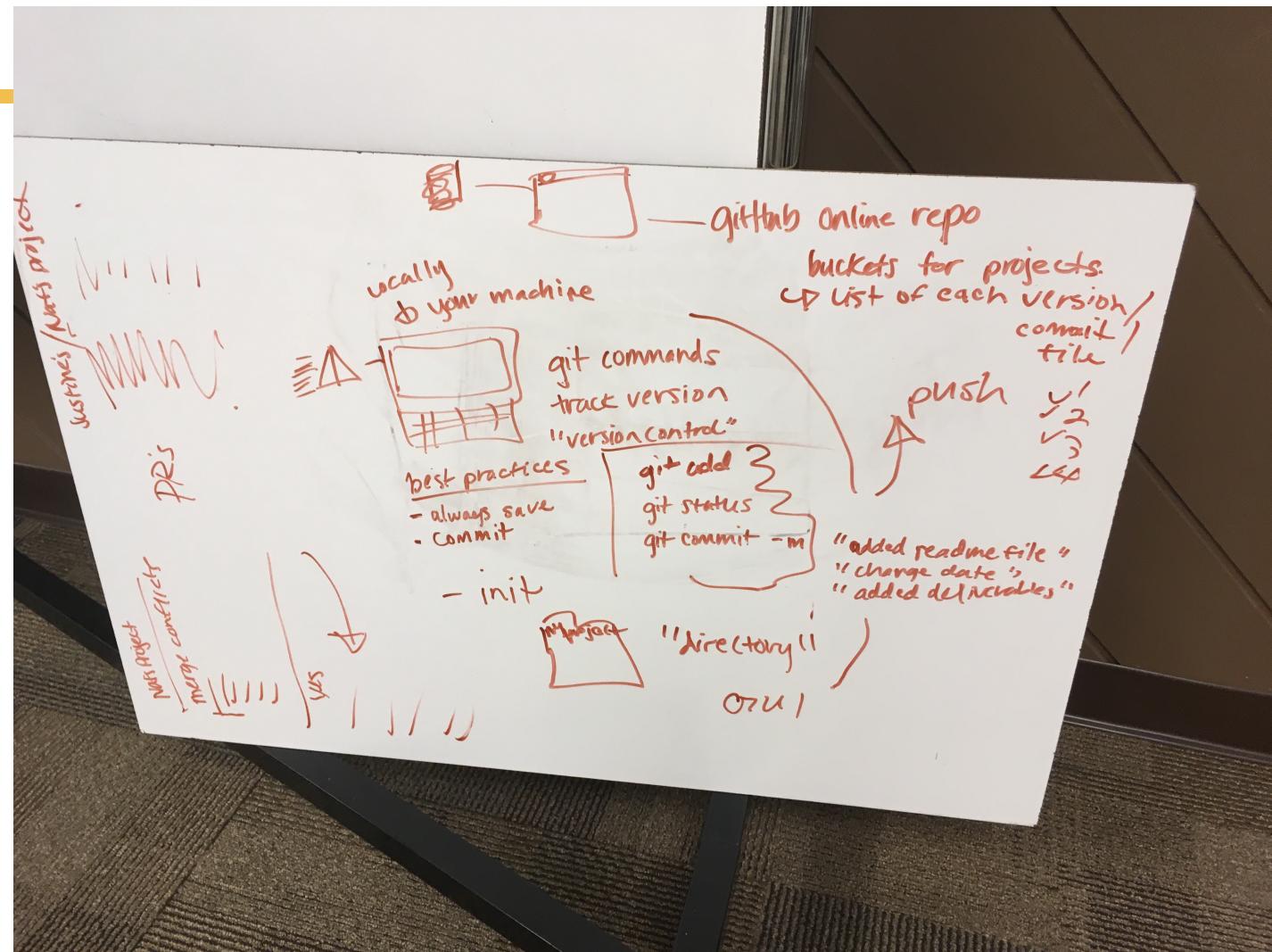


He showed the addition of origin of purchase being added as what I'd call a field in this data file that I guess is going into Console/Comet, but it's still unclear if that's accessible for us to pull ourselves. If I want to track an event, could PBSC make me an API key so I can access it myself? Is it easier or harder for them to readily pull the report for me/Secure Bike Share on a monthly basis?



Looks comprehensible here, but I get lost in the folders and instructions on my computer

Trying to comprehend why we even need many different versions of things. DevLeague git hub is public space to share amongst ourselves, and I can only put things into there off of my local computer through my own github page



Membership/Subscriber Data Collected at Purchase from Website Server -> PBSC Server or Phone App -> Website -> PBSC Server

Ride Data Collected at Start Station -> PBSC Server -> Console -> Comet

Ride Data Collected at End Station -> PBSC Server -> Console -> Comet

Customer ID	First Name	Last Name	User Type	Product ID	Zip Code	Product Name	Start Time	Start Station	Start Station Name	End Time	End Station	End Station Name	Trip Duration (sec)	Duration
15039	Sheuli	M	Subscriber	2	96814	\$15 Monthly	0:17:29	73	103 Downtown/CT - Fort & Hotel	0:37:27	49	235 Kak/AlaMo - Amana & Makaloa	1198	0:19:58
24075	Hunter	K	Subscriber	5	96815	Free Spirit Pass	0:34:09	26	330 Waikiki - Liliuokalani & Mounta	0:37:44	28	313 Waikiki - Kalaimoku & Ala Wai	215	0:03:35
17664	Justine	E	Subscriber	2	96814	\$15 Monthly	0:35:13	79	460 McCully/Moiliili - Kapiolani & M	0:43:03	50	300 Waikiki - Ala Wai & Lipeepee	470	0:07:50
10570	Olina	F	Subscriber	2	96814	\$15 Monthly	0:40:47	50	300 Waikiki - Ala Wai & Lipeepee	0:48:35	48	234 Kak/AlaMo - Rycroft & Ahana	468	0:07:48
24034	Natalie	N	Subscriber	2	96826	\$15 Monthly	0:55:15	103	332 Waikiki - Kalakaua & Ohua	1:10:16	46	456 McCully/Moiliili - Date & Paani	901	0:15:01
24329	Ranjini	M	Customer	4		Single Ride	0:57:07	18	314 Waikiki - Saratoga & Kalakaua	1:06:37	51	241 Kak/AlaMo - Ala Moana Park & Atkinson	570	0:09:30
24329	Victor	R	Customer	4		Single Ride	0:57:26	18	314 Waikiki - Saratoga & Kalakaua	1:06:50	53	243 Kak/AlaMo - Ala Moana Park & Atkinson	564	0:09:24
1034	Tori	P	Customer	3	96826	Free Spirit	0:51:48	69	105 Downtown/CT - S.Kukui & Fort	0:55:29	69	105 Downtown/CT - S.Kukui & Fort	221	0:03:41
5486	Michael	R	Customer	3	96815	Free Spirit	0:21:00	89	328 Waikiki - Kalakaua & Uluniu	0:26:12	12	504 DiamondH/Kap - Kapahulu & Castle	312	0:05:12
15094	Ben	D	Customer	3	96712	Free Spirit	0:21:00	79	460 McCully/Moiliili - Kapiolani & M	0:34:41	28	313 Waikiki - Kalaimoku & Ala Wai	821	0:13:41
10602	Justin	O	Customer	3	96815	Free Spirit	0:13:05	30	321 Waikiki - Seaside & Kalakaua	0:21:12	42	301 Waikiki - Hobron & Kaioo	487	0:08:07

Casual/Customer Data Collected at Purchase Kiosk -> PBSC Server or Phone App -> Website -> PBSC Server

Only file formats I've seen are CSV

# Data Collection Wish List

Big dreams: hierarchical data?

## Trips by Product Type

When they ride

Average trip length by product type

Average frequency by product type

Trips by area

Trips origin/ending by area

Popular routes (A-B)

Popular routes (A-A)

Trips per person

Impact of rain to rides

Impact of heat to rides

Impact of events to rides

Remaining balance in FS minutes

Total minutes used in FS minutes (tourists)

Average bikes checked out in 1 transaction

Person, zipcode, start station, end station

External data sets,  
might be easier to  
use with what I  
already get from  
Secure

This is a line item for each customer in Console



## In Conclusion

As always, there are still plenty more things we could study in the dataset. Bad weather probably affects cycling speeds, so we could take that into account when measuring speeds and Google Maps time estimates.

Ben Wellington at I Quant NY did some [demographic analysis by station](#), it might be interesting to see how that has evolved over time.

I wonder about modeling ridership at the individual station level, especially as stations are added in the future. Adding a new station is liable to affect ridership at existing stations—and it's not even clear whether positively or negatively. A new station might cannibalize trips from other nearby stations, which wouldn't increase total ridership by very much. But it's also possible that a new station could have a synergistic effect with an existing station: imagine a scenario where a neighborhood with bad subway access gets a Citi Bike station, then an existing station located near the closest subway might see a surge in usage.

There are also probably plenty of analyses that could be done comparing Citi Bike data with the [taxi and Uber data](#): what neighborhoods have the highest and lowest ratios of Citi Bike rides compared to taxi trips? And are there any commutes where it's faster to take a Citi Bike than a taxi during rush hour traffic? Alas, these will have to wait for another time...

## GitHub

There are scripts to download, process, and analyze the data in the [nyc-citibike-data repository](#). A csv of the [raw data for the weather analysis](#) (daily trip totals plus weather data) is included in the repo, in case you don't want to download all of the

A screenshot of a web browser window showing a list of tabs at the top. The tabs include: Desk, READ, REAL, devle, BigD, j-esp, BDA, Wha, Bik, TA Tal, Todd, Com, Justinfre..., and toddwschneider.com/posts/a-tale-of-twenty-two-million-citi-bikes-analyzing-the-nyc-bike-share-system/. Below the tabs, the address bar shows the URL: toddwschneider.com/posts/a-tale-of-twenty-two-million-citi-bikes-analyzing-the-nyc-bike-share-system/. The main content area displays the text from the 'In Conclusion' section above. At the bottom of the browser window, there is a GitHub repository page titled 'NYC Citi Bike Data' with instructions and code snippets for setting up the database.

## NYC Citi Bike Data

Code in support of this post: [A Tale of Twenty-Two Million Citi Bikes: Analyzing the NYC Bike Share Sys](#)

This repo provides scripts to download, process, and analyze data for NYC's [Citi Bike system data](#). The [PostgreSQL](#) database, uses [PostGIS](#) for spatial calculations, and [R](#) for data analysis.

Pretty much a copy of the [taxi/Uber data repo](#), at some point the Citi Bike, taxi, and Uber datasets were combined into a single unified NYC transit database...

### Instructions

1. Install [PostgreSQL](#) and [PostGIS](#)

Both are available via [Homebrew](#) on Mac OS X

2. Download raw taxi data

```
./download_raw_data.sh
```

3. Initialize database and set up schema

```
./initialize_database.sh
```

4. Import taxi data into database and map to census tracts

```
./import_trips.sh
```

5. Analysis

Additional Postgres and R scripts for analysis are in the `analysis/` folder

### Other data sources

These are bundled with the repository, so no need to download separately, but:

A screenshot of a Windows taskbar at the bottom of the screen. It features a search bar with the text 'Type here to search'. To the right of the search bar are several pinned icons for various applications, including File Explorer, Google Chrome, Microsoft Edge, Spotify, and others. On the far right, there is a system tray with icons for battery status, signal strength, and a notification bubble indicating four unread messages. The taskbar is dark-themed.



# Fin