Rails in real-time

Hi 👋

I'm Anton

Apologies

Story about startup

Monopoly of public transport

then Uber comes along

(and others)

and government acts

New business opportunity

While in California

2 years later

Ford paid more than $65 million for shuttle-van startup Chariot

Matt Rosoff and **Biz Carson**   Sep. 12, 2016, 11:54 PM   🔥 13,258

| FACEBOOK | LINKEDIN | TWITTER | EMAIL | PRINT |

Last Friday, Ford bought shuttle-van startup Chariot in an all-cash deal. But the price wasn't disclosed, so it wasn't clear whether this was a case of a distressed startup selling for a bargain-basement price, or Ford paying a solid price for
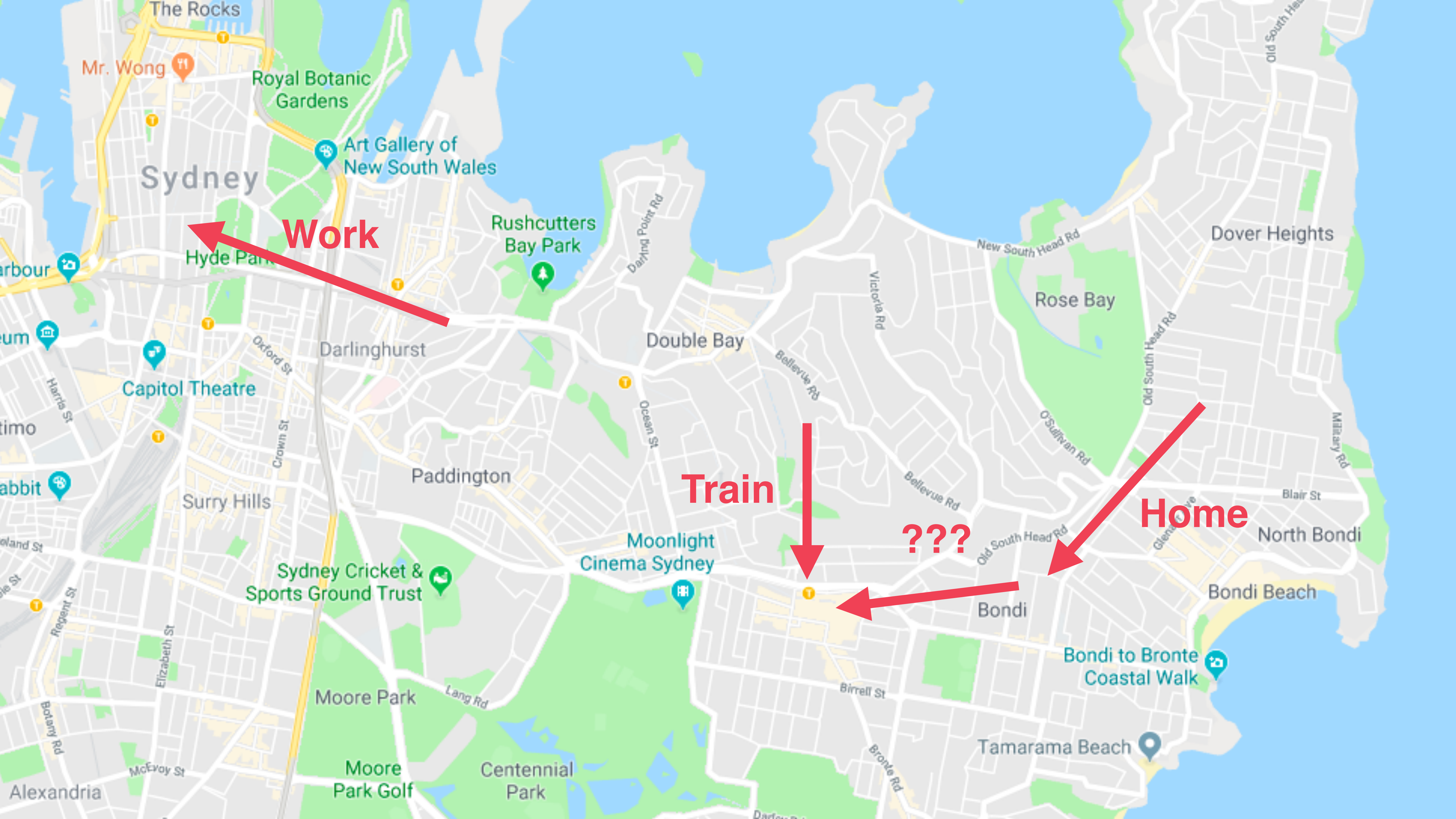
What is it?

Chariot is a commuter shuttle service.

The company's mobile-phone application allows passengers to ride a shuttle between home and work during commuting hours.

November 2016

That's when I was hired to build that in AU
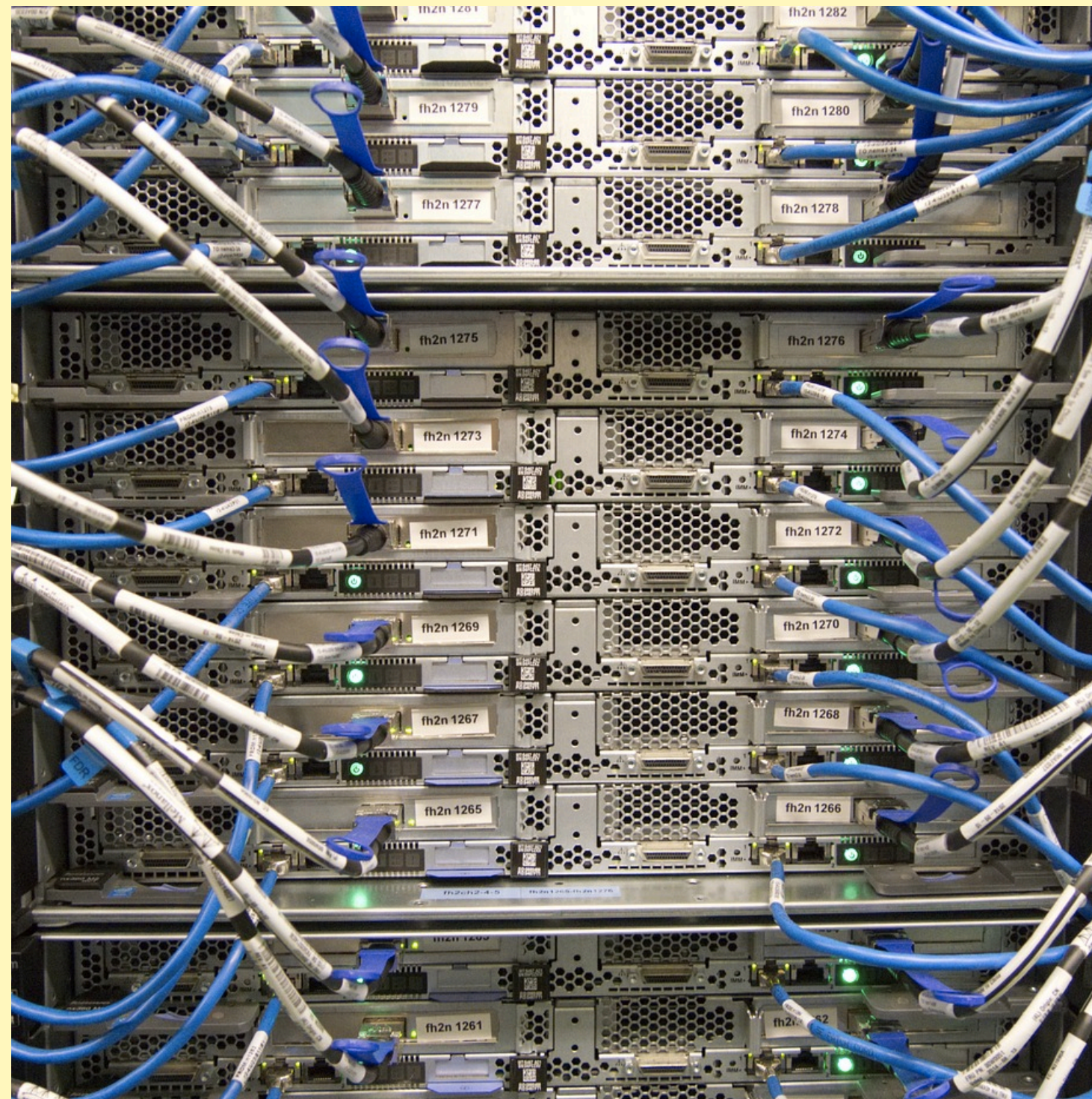
The problem

Uber like app

- Book a seat on the bus

- See bus position in real time

- Once boarded, show the estimated arrival time

The only difference from Uber app: you need to choose pick up and drop off stops along the route.

The Game Plan

Passenger app

Server

Driver app

Passenger app

- We chose Ionic Framework 2 (Angular)

  - I had experience with it before

  - Hybrid app (Android and iOS)

  - Great for prototyping
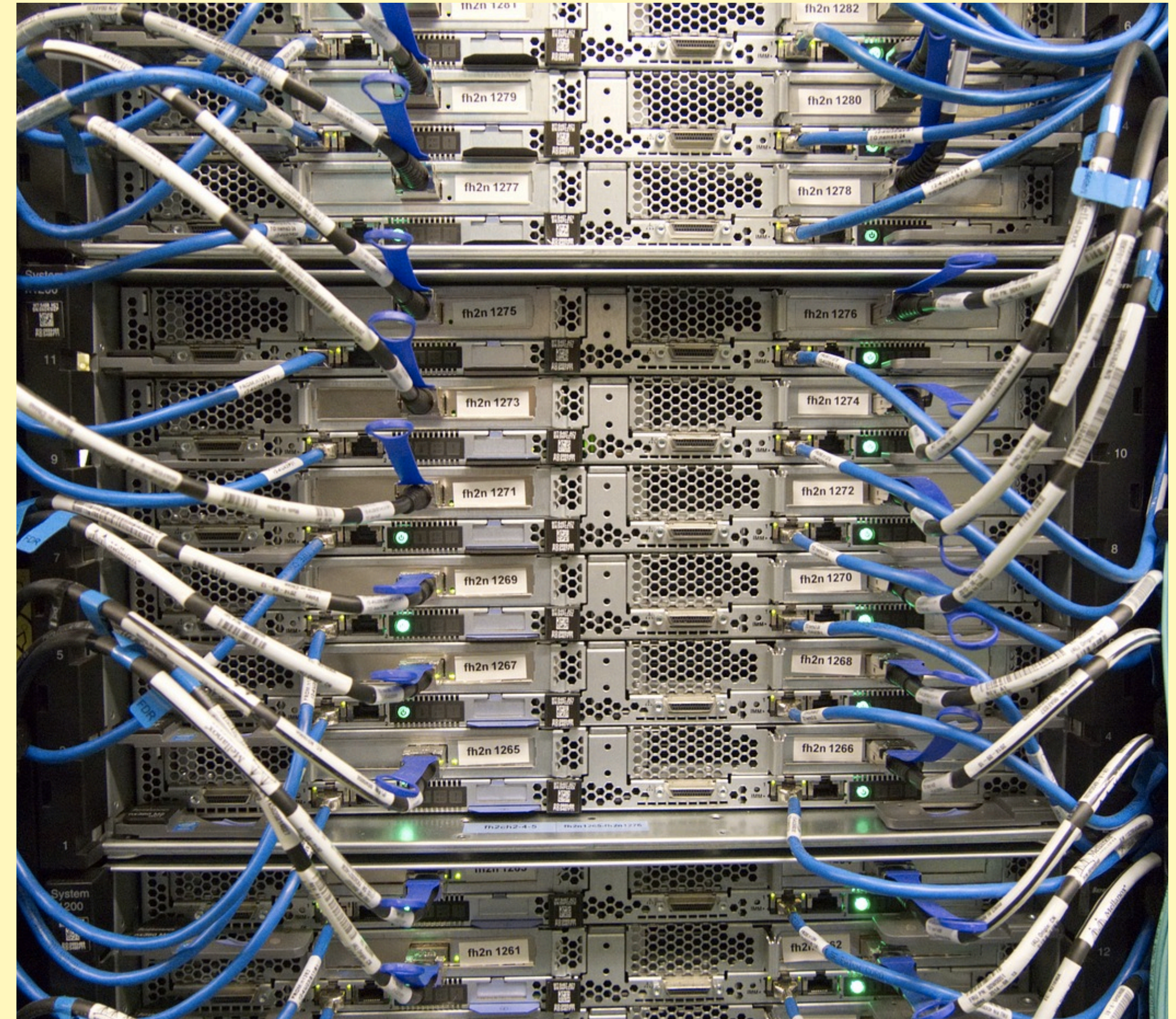
  - Good ecosystem

Driver app

- Sends real time position

- Shows map, stop times, passengers

- Options:

  - Separate mobile app

  - Integrate into Passenger app

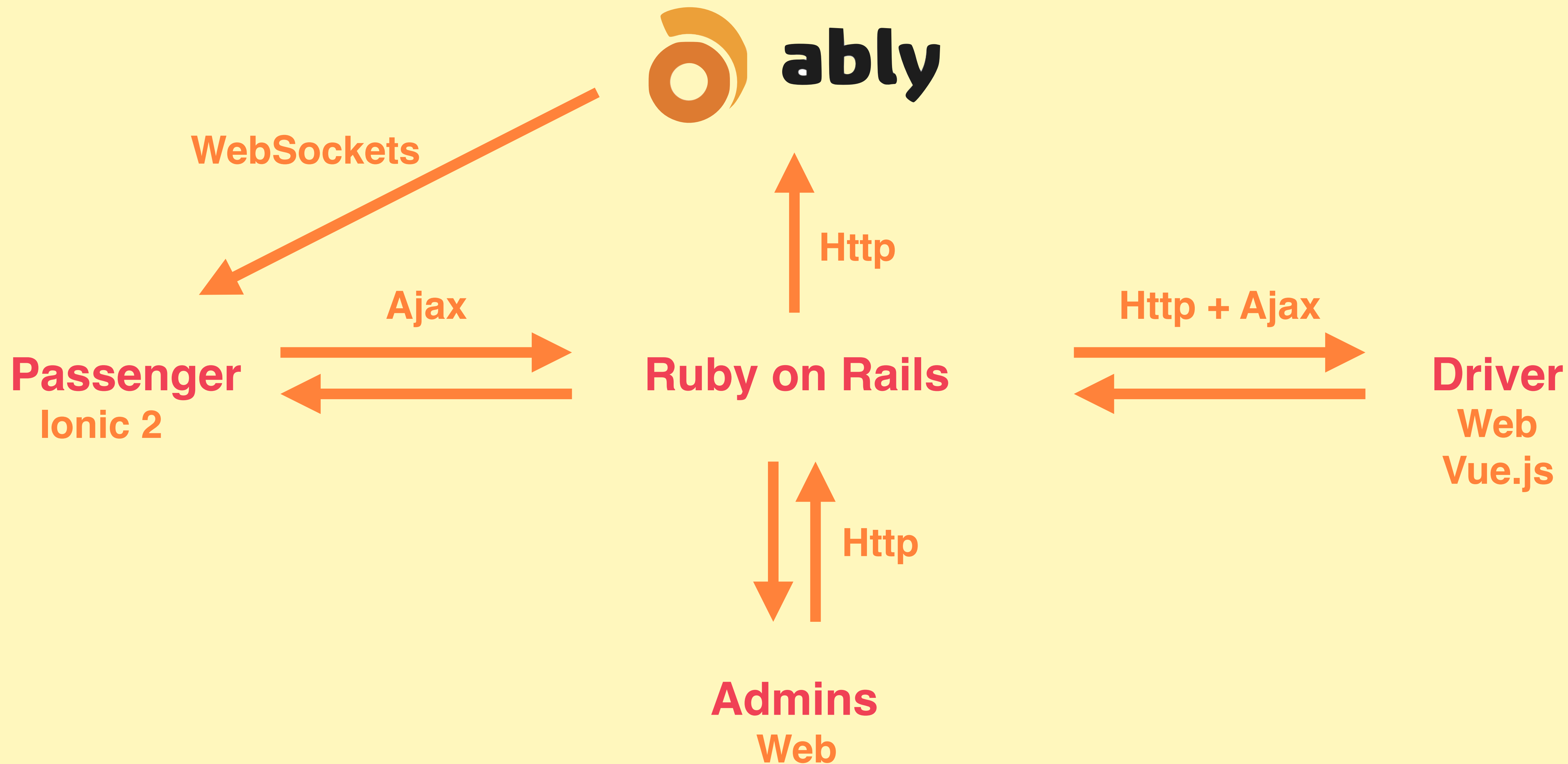 - No mobile app (web page with Vue.js)

Server

- Real time data

  - Elixir is hot at the time and fits perfectly

- My specialty is Ruby & Rails

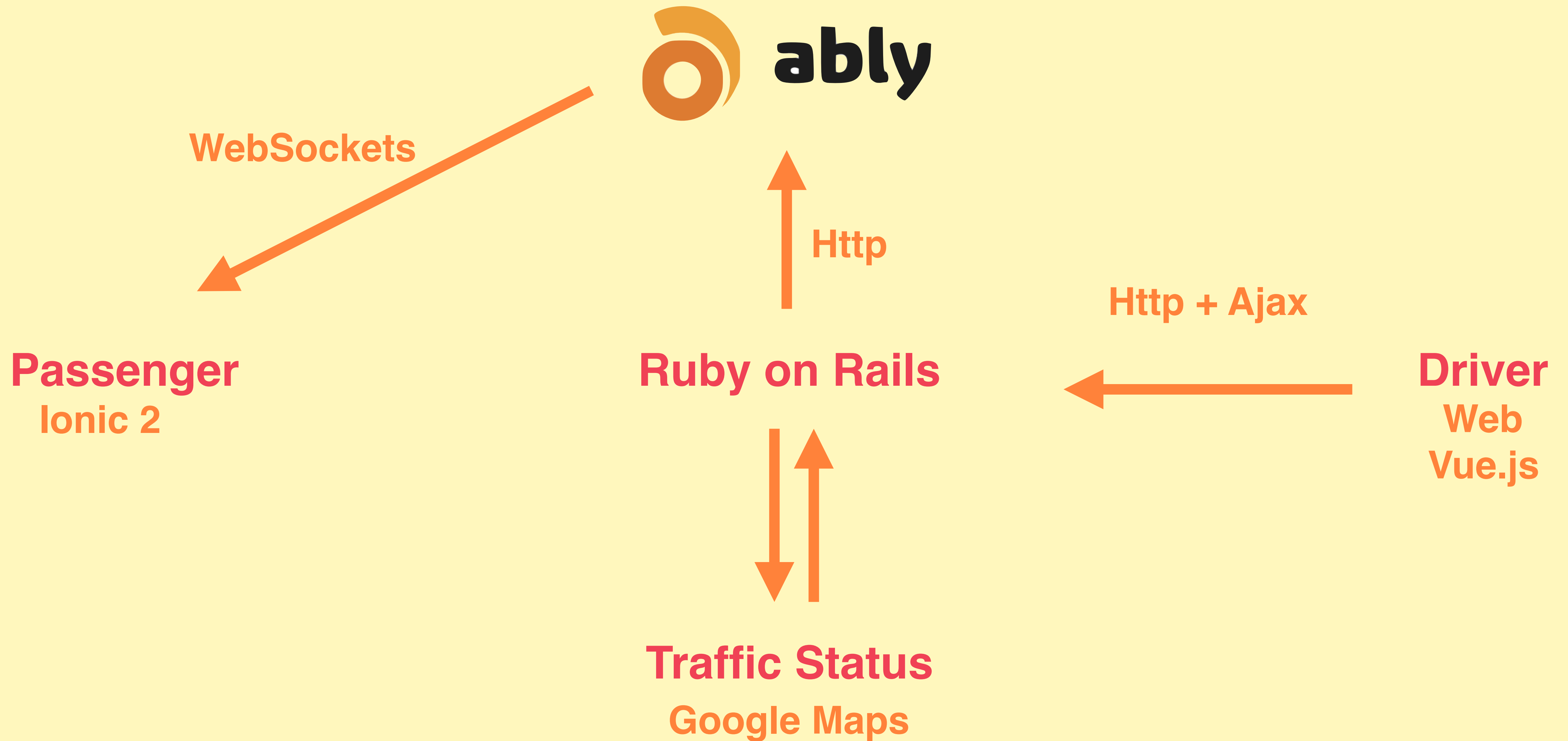- Use correct technology or deliver the product

- We chose Rails

System overview

Action Cable?

- It's November 2016, remember?

  - Action Cable is 4 months old

- Would I use it today?

  - Probably

Real time updates

Rails with
Async Architecture

My location is X

Driver → **Controller**

Update bus location

→ **Event**

Broadcast location

→ **Event**

Save location

↓

Broadcast

↓

Check traffic on Google

Cron → **Event**

Save traffic data

→ **Redis**

ably.io

Check stop reached

↓

Broadcast

↑

Event

Broadcast schedule
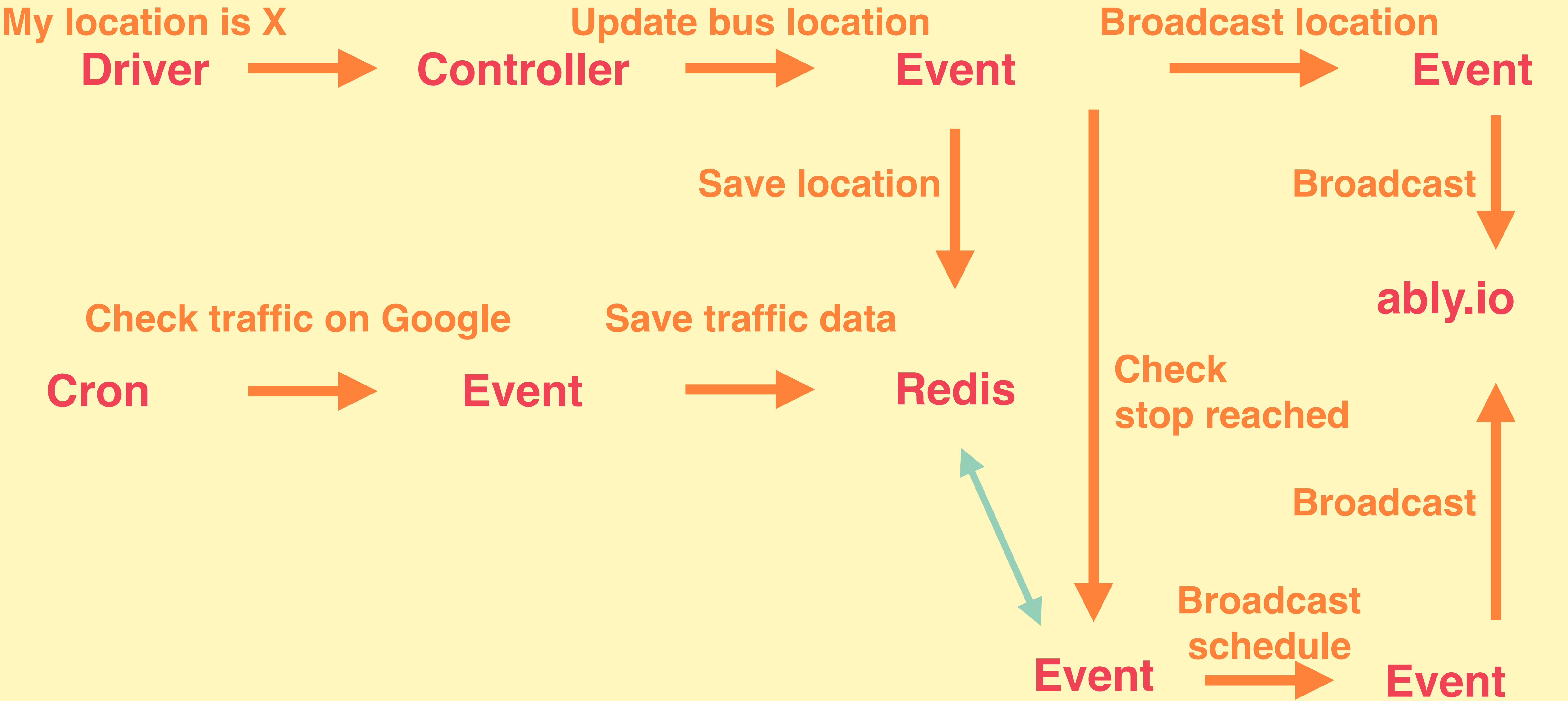
→ **Event**

Event?

- Event is a background Job

  - Powered by **sidekiq** gem

- Cron: **sidekiq-cron** gem

- Redis: **redis-objects** gem

- 0 workers / jobs

Service Objects

- Each event is a service object

- Each service object can be sync / async

- Gems

  - Backbone: **active_interaction**

  - Async: **active_interaction-extras**

- Swiss army knife

  - Form objects

  - Service objects

  - Workers

The challenges

GPS issues

GPS issues

- Browser sends GPS location only if tab is active

- GPS signal is not accurate within city

- Balance between accuracy and real-time

- Our app was more real-time than Uber

Predicting the future

Goal: better than 5% accuracy
~1 minute for 30 minutes prediction

Uber is easy

Morning shift

- One way 14 minutes A to B

  - 9 stops

- Return 10 minutes B to A

  - 8 stops

- Loop 24 minutes

- 7 loops per 3 hour shift

  - Total 119 stops

- 6:00am A to B

- 6:14am B to A **+ 5min**

- 6:24am A to B **+ 5min**

- 6:48am B to A **+ 5min**

- 6:58am A to B **+ 5min**

- … etc up to 9am

Traffic data

- Use Google Maps

- Query travel time for each of 119 stops

- Refresh every 3 minutes

- Store in Redis

- Problem: Google traffic estimates are delayed

Does it scale?
Yes

Does it scale?

- Heavy lifting done by ably.io

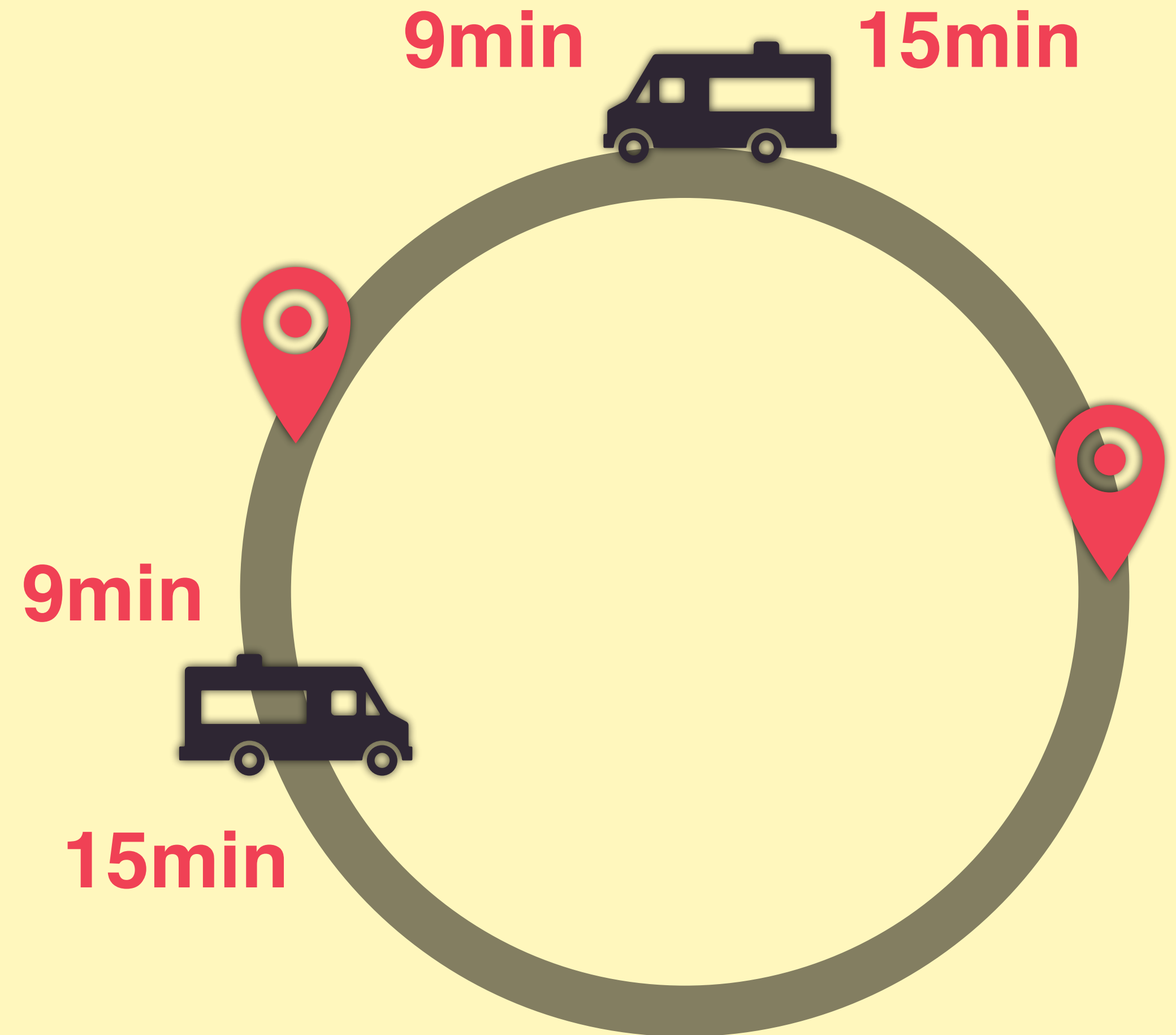- All you have to deal with is few api requests per user

Does it scale with buses?

Solution

- Show driver the time distance for the bus in front and behind

- Now driver can slow down if he is too fast

9min **15min**

9min

15min

All done
but still not good enough

Still not good enough

- Dynamic schedule which adjusts to road conditions

- Google Maps traffic

- Real time bus location

- Drivers that balance the schedule

- Estimates are still not accurate enough

  - Solution: Add buffer time for each route. E.g. +15%

The A-team

- Developer* (me) - full time

- Designer* - part time

- Business guy - part time

- Operations guy - part time

  first timer

- Private trial in 2.5 months

- Launch in 3.5 months

- Closed after 5 months

  - due to business reasons

Conclusion

Thank you