OTP Transit Index and TriMet

Data and functional gaps in the Transit Index regarding TriMet data

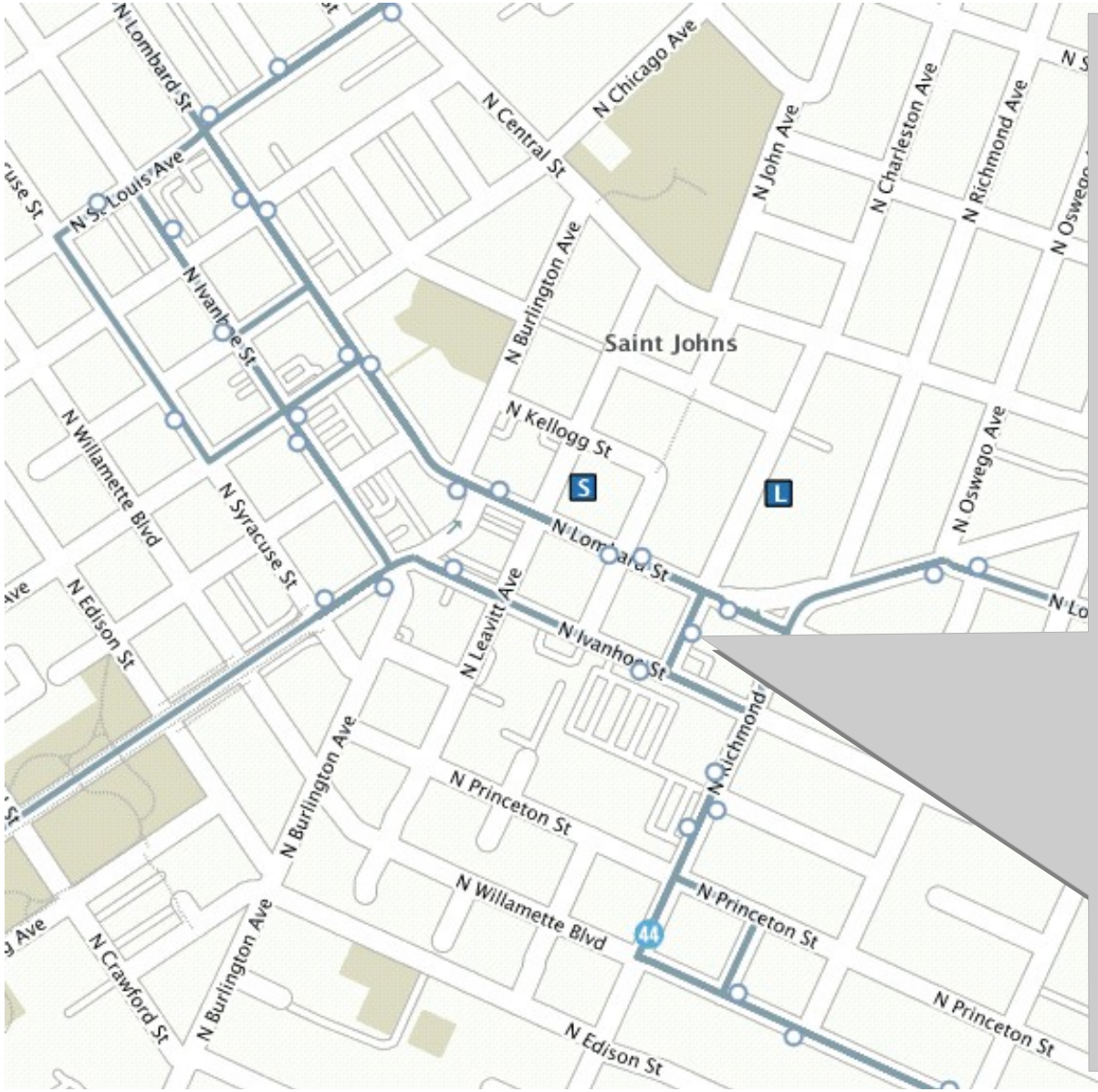
November 2016

issue #1: current stops

Things to look at:

- On ride.trimet.org stops are only shown that are currently ***active***
- Look at the circles between this page and the next, and notice how many more stops show up on call.trimet.org

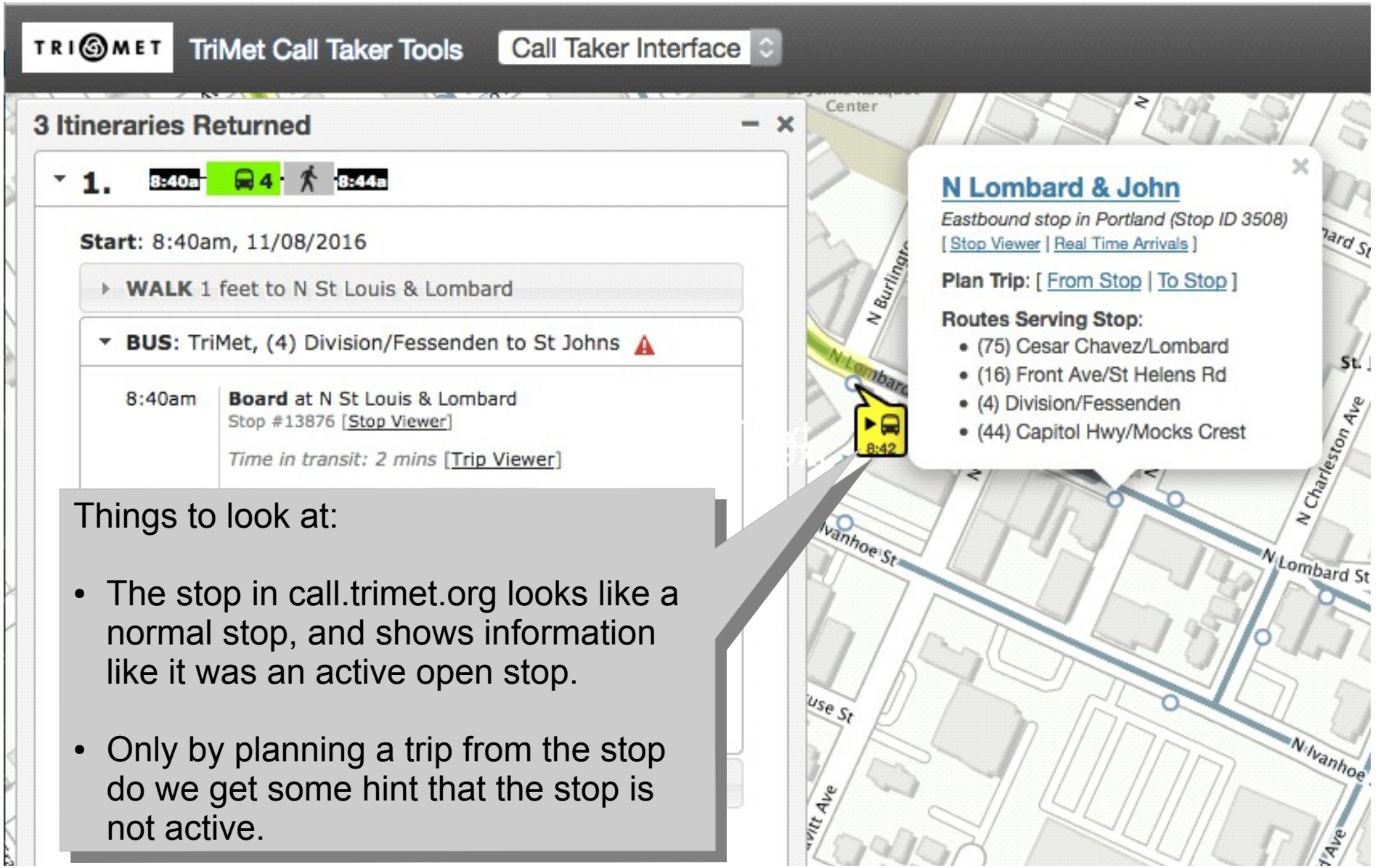
issue #1: current stops



Things to look at:

- The call.trimet.org system has a stop overlay like ride.trimet.org
- Problem is there's no filtering of stops that appear in TriMet's GTFS.zip file.
- Often stops in the GTFS are not active at certain points in the schedule.
- Determining an effective schedule period requires looking at all the trips in the GTFS, and determining which stops have service … for this reason, we choose to go the gtfsdb route for our pages

issue #1: current stops



Things to look at:

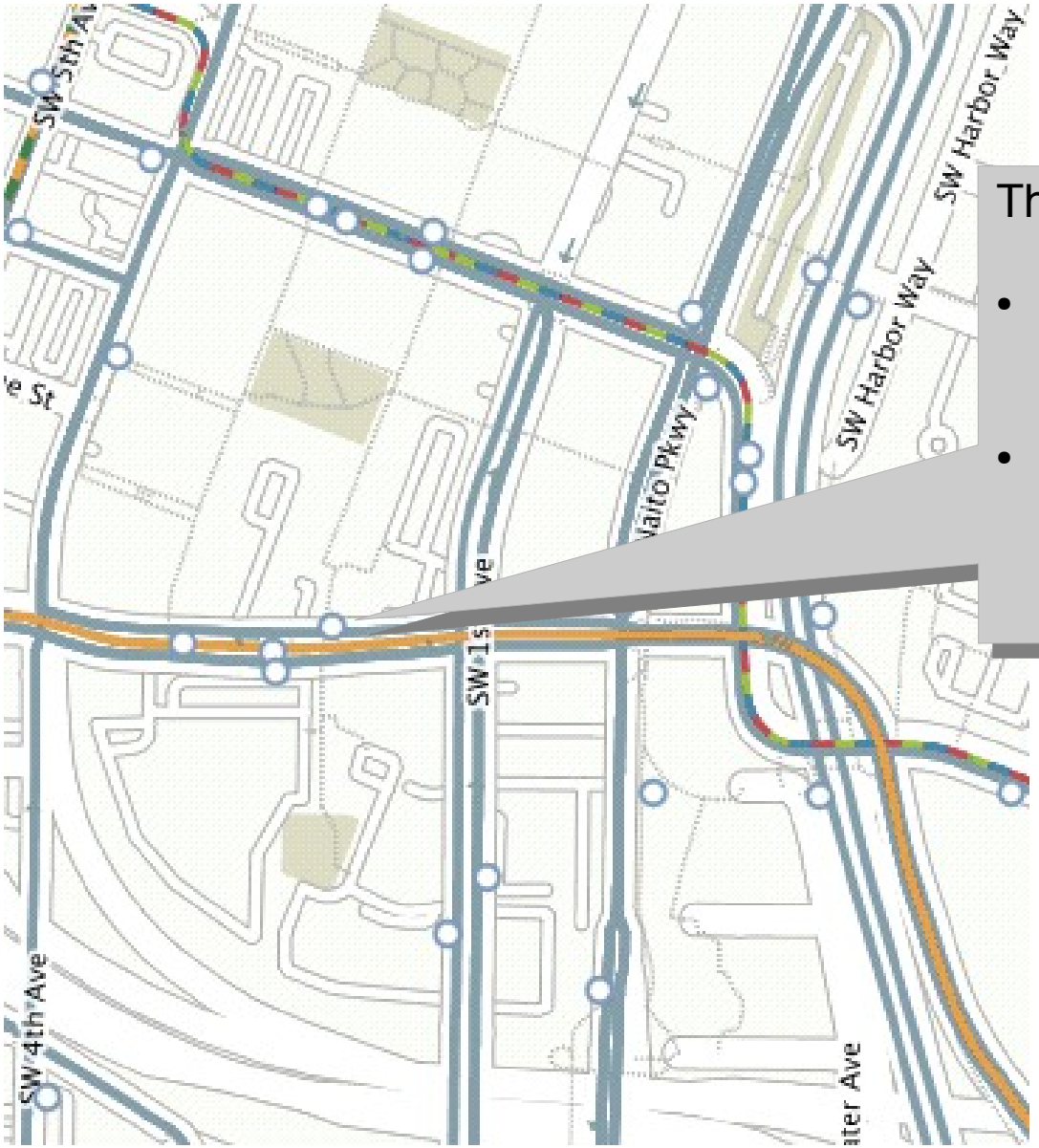
- The stop in call.trimet.org looks like a normal stop, and shows information like it was an active open stop.
- Only by planning a trip from the stop do we get some hint that the stop is not active.

issue #2: stops types

Things to look at:

- On ride.trimet.org, we have 5 different icons for stops.
- Here you see blue bus icons, small black streetcar stops and larger black MAX icons

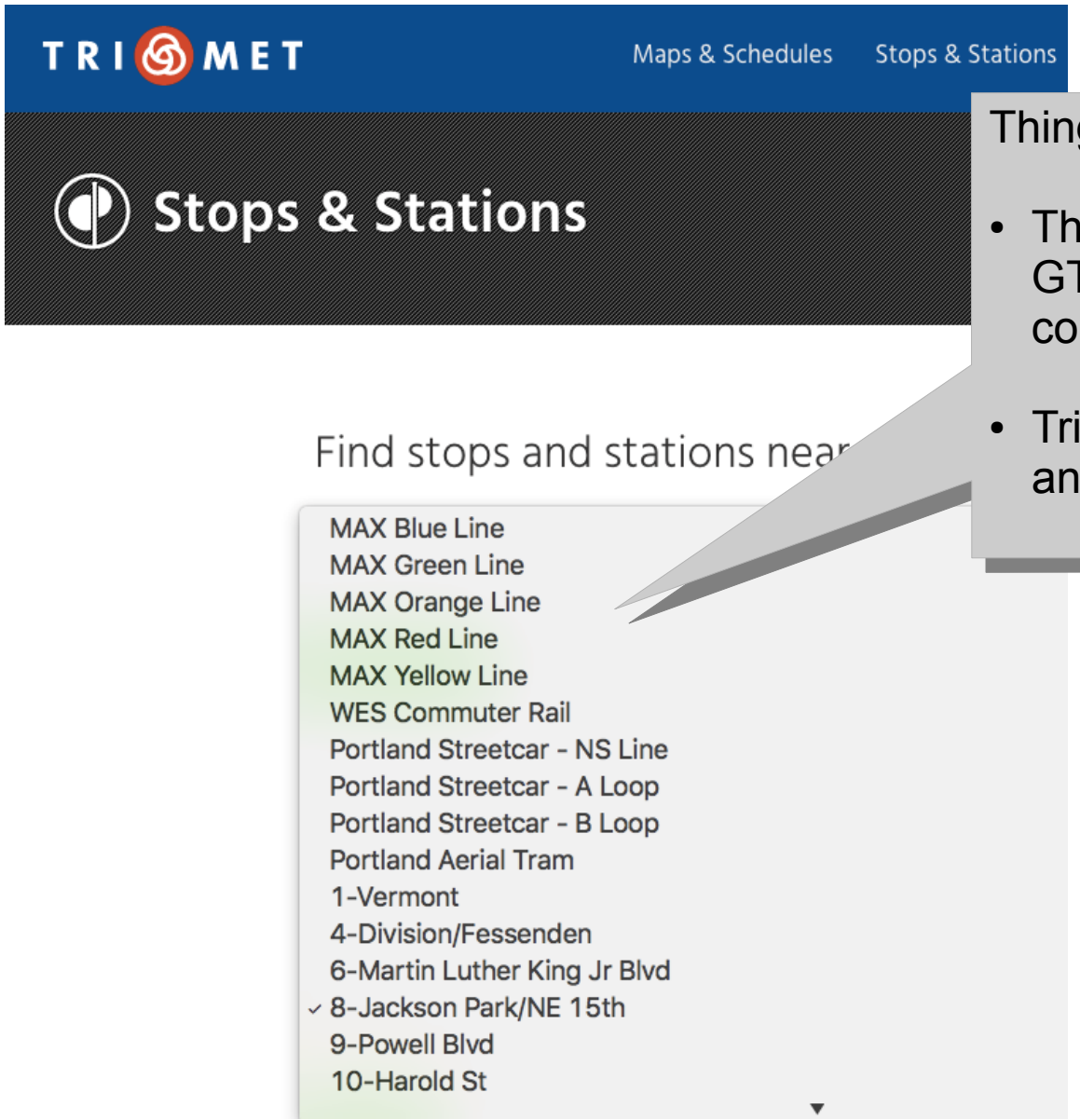
issue #2: stops types



Things to look at:

- Within the OTP transit index, there is no concept of stop (mode) type.
- Thus in call.trimet.org, there's no differentiation between the different rail and bus modes servicing stops.

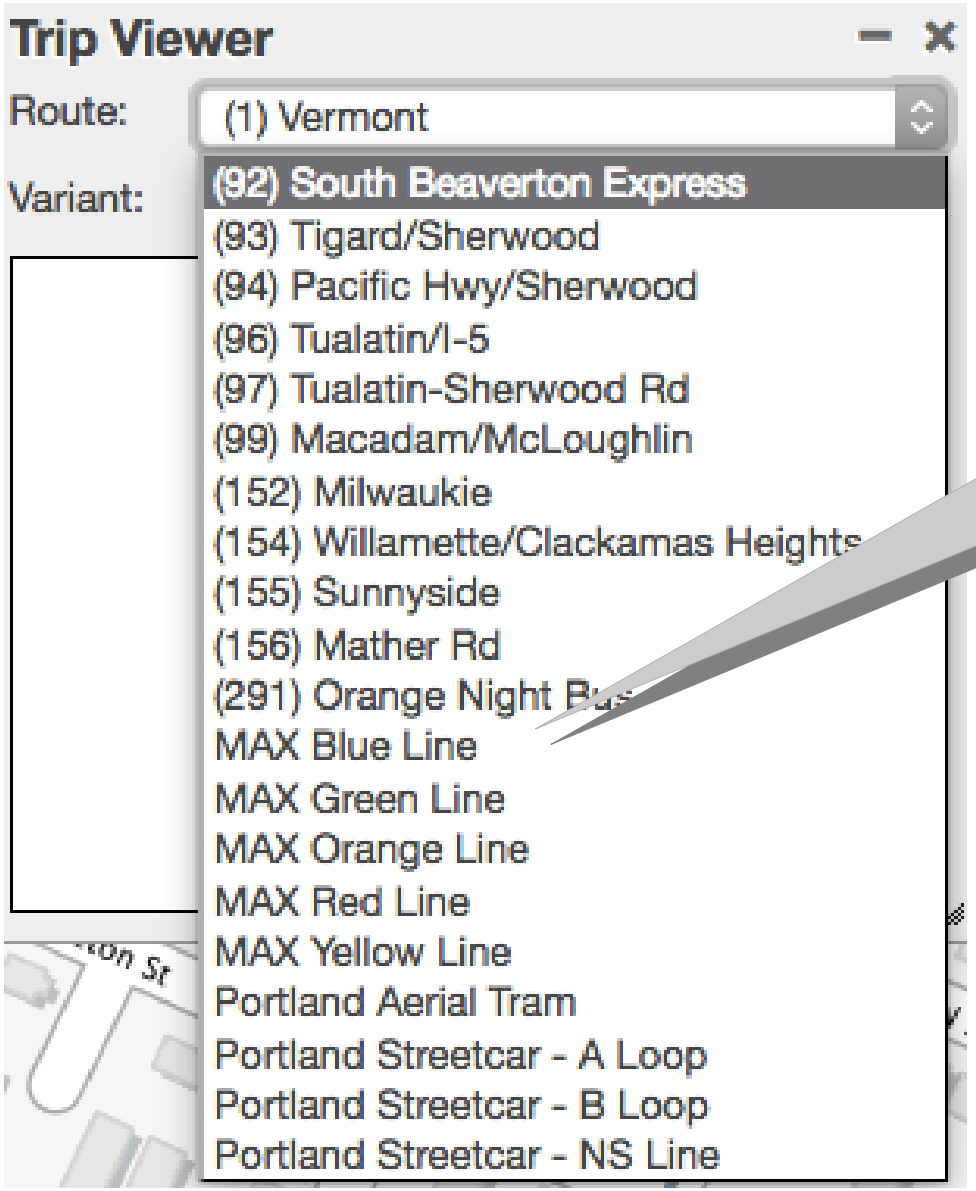
issue #3: route sort order



Things to look at:

- There's a proposed extension in GTFS, where a route sort order column is added to routes.txt
- TriMet supplies the route sort order, and gtfsdb reads and returns this info

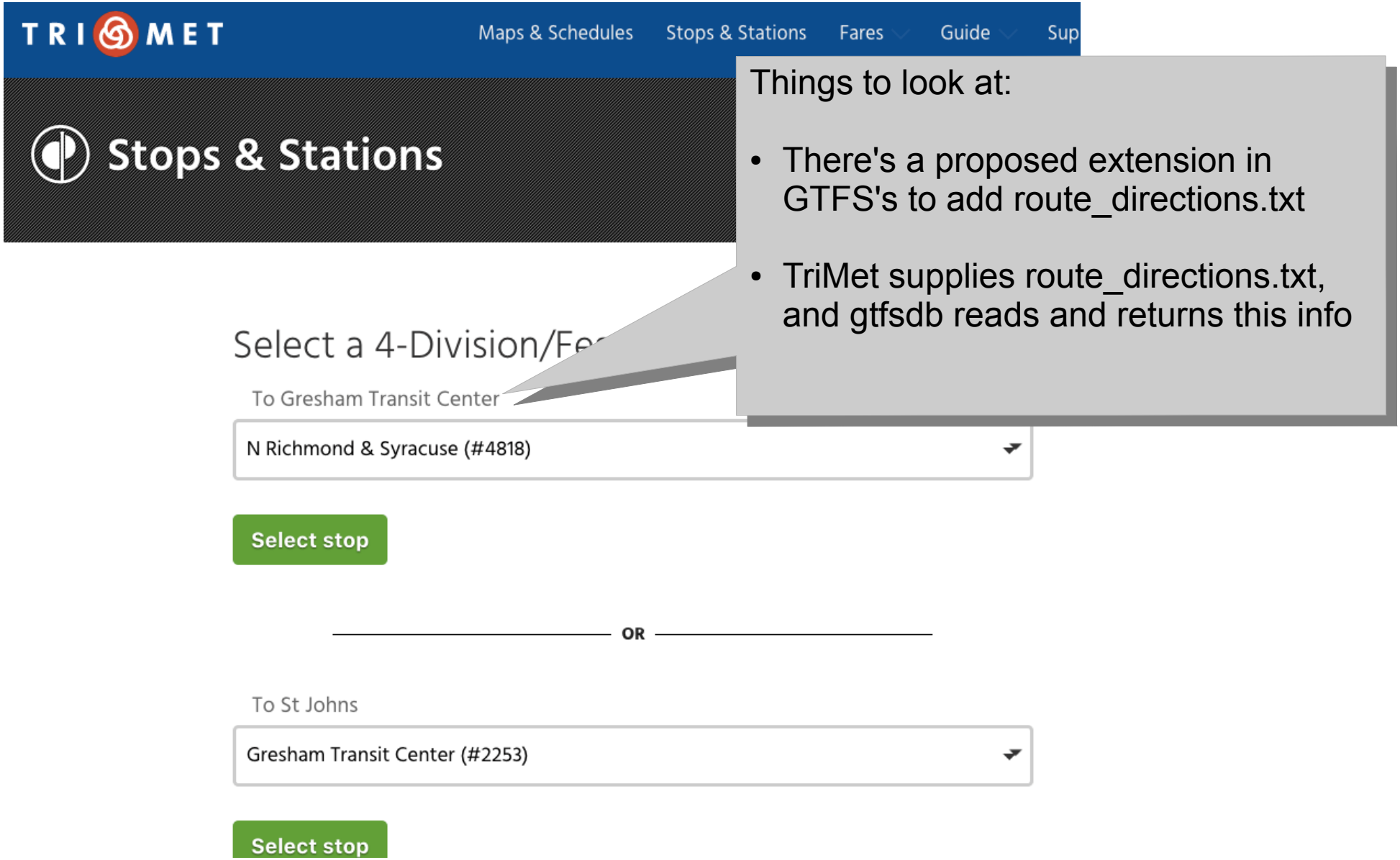
issue #3: route sort order



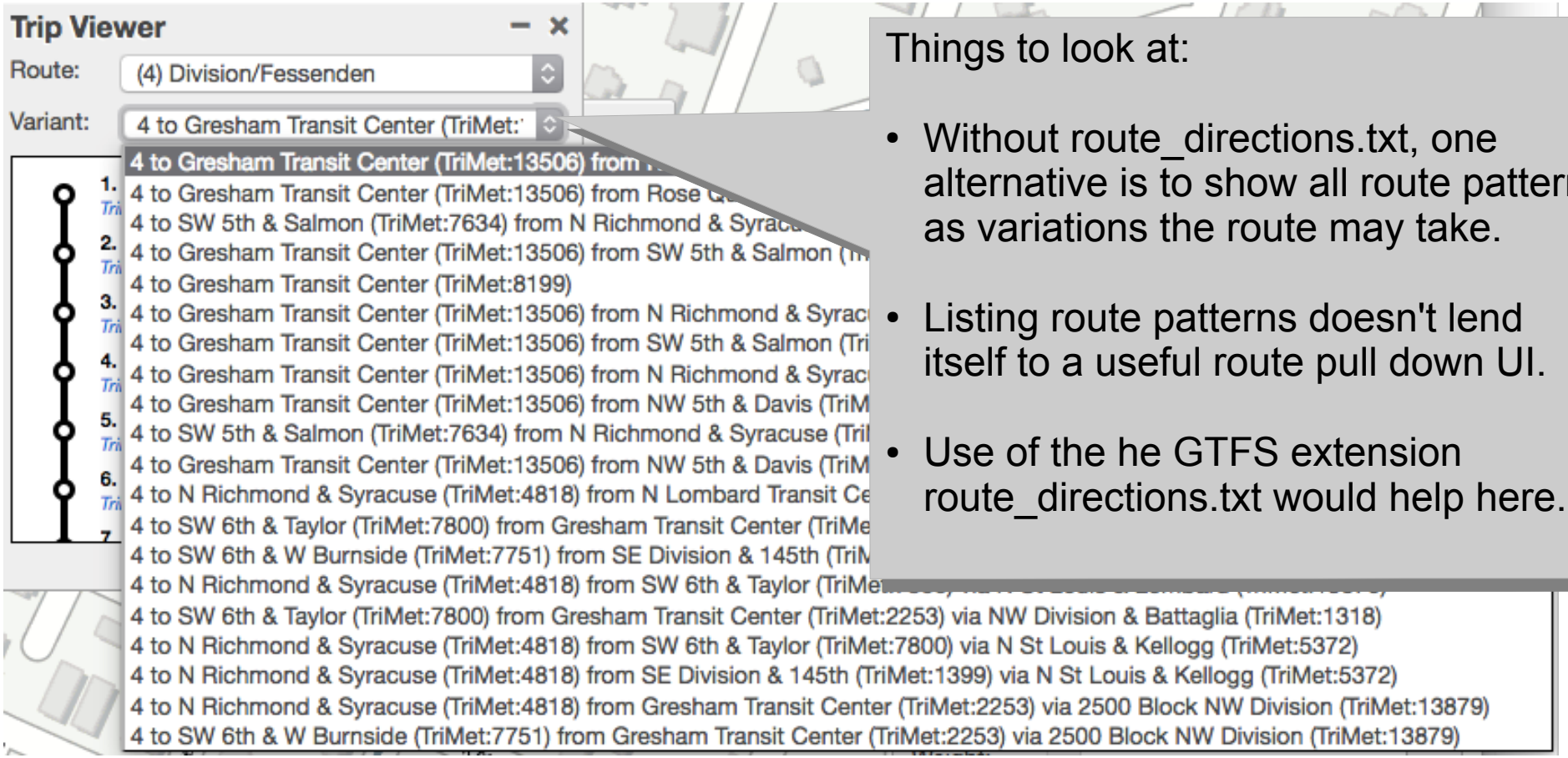
Things to look at:

- The OTP transit index (and OBA gtfs libs) are not reading route sort order

issue #4: route direction



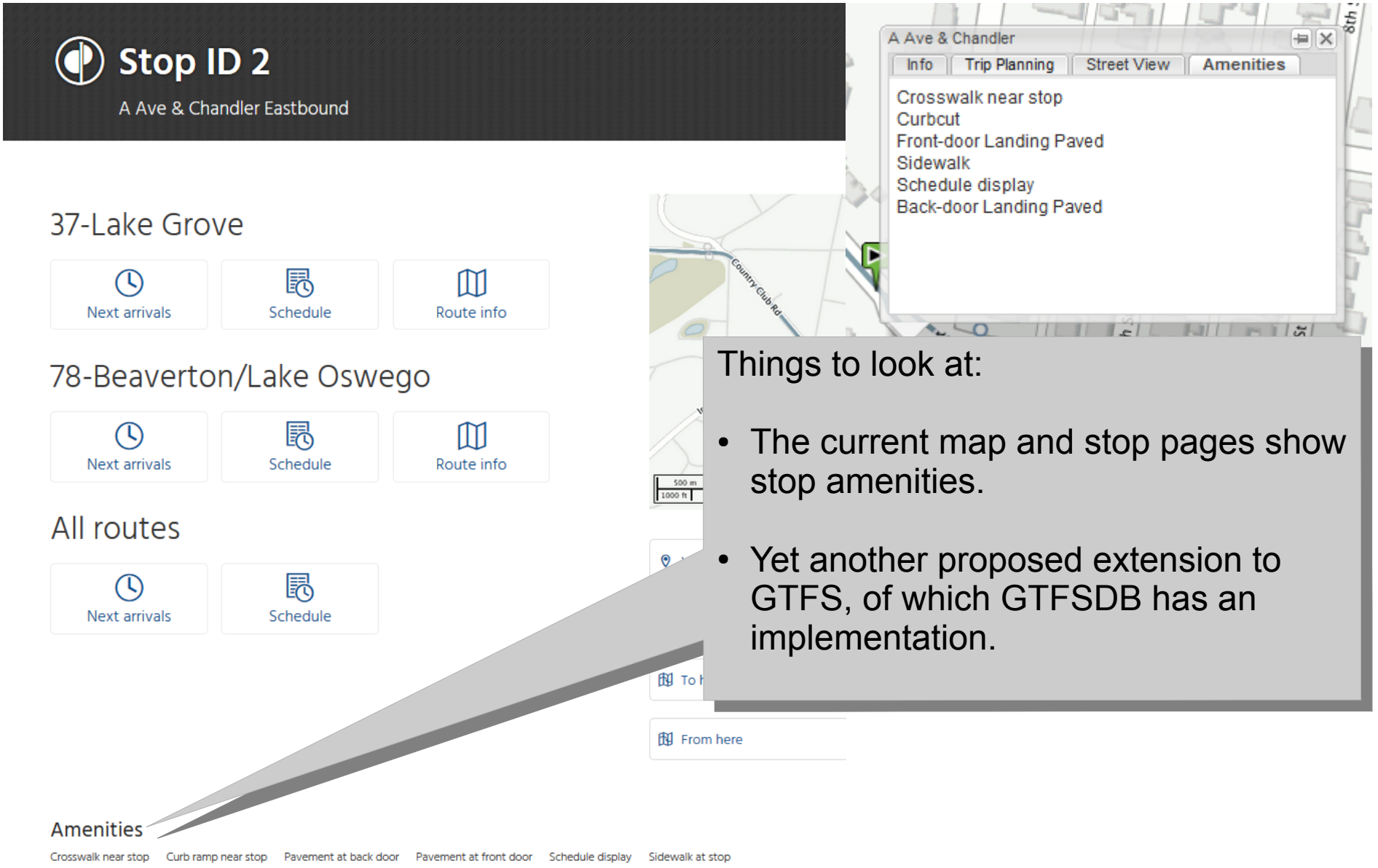
issue #4: route direction



Things to look at:

- Without route_directions.txt, one alternative is to show all route patterns as variations the route may take.
- Listing route patterns doesn't lend itself to a useful route pull down UI.
- Use of the he GTFS extension route_directions.txt would help here.

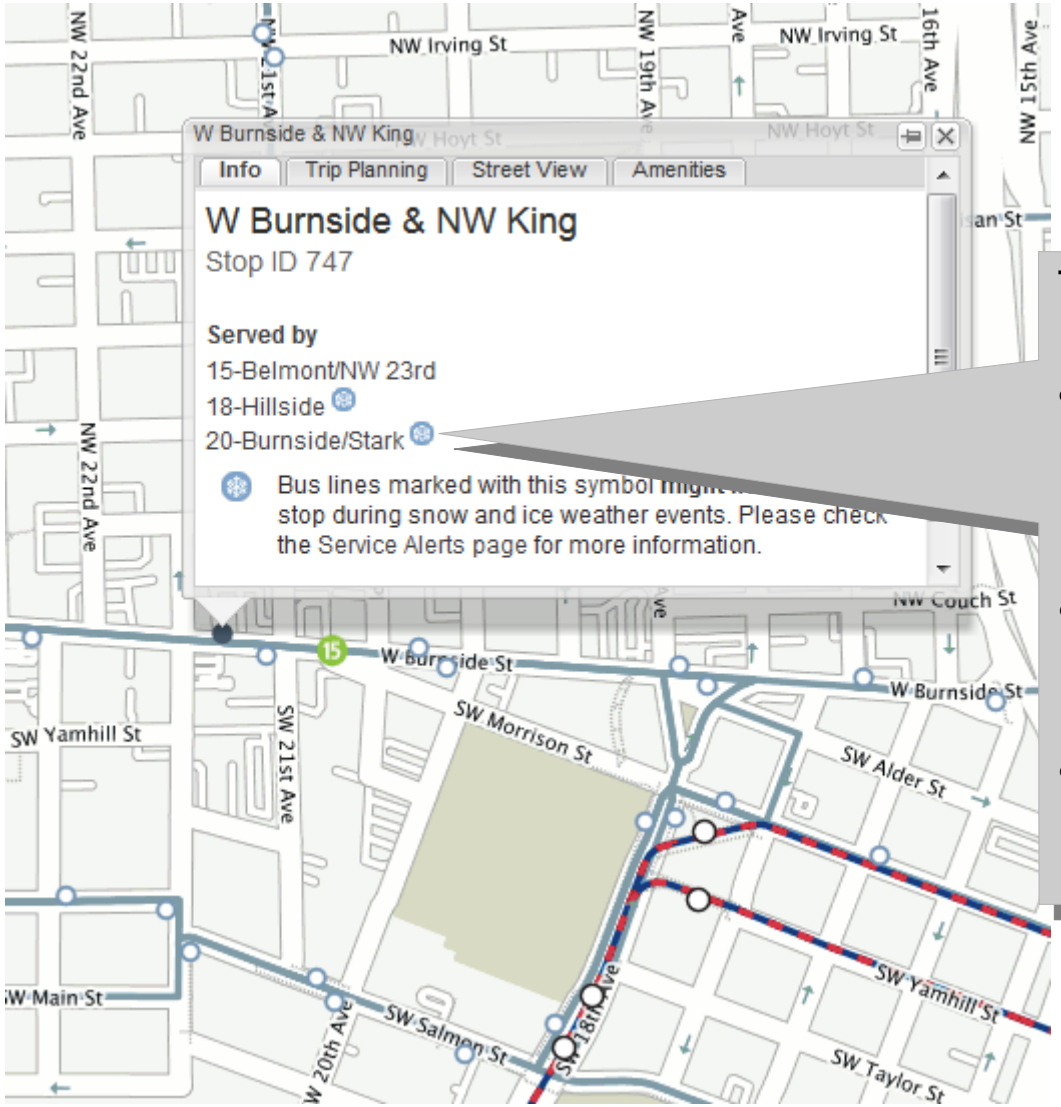
issue #5: stop amenities



Things to look at:

- The current map and stop pages show stop amenities.
- Yet another proposed extension to GTFS, of which GTFSDB has an implementation.

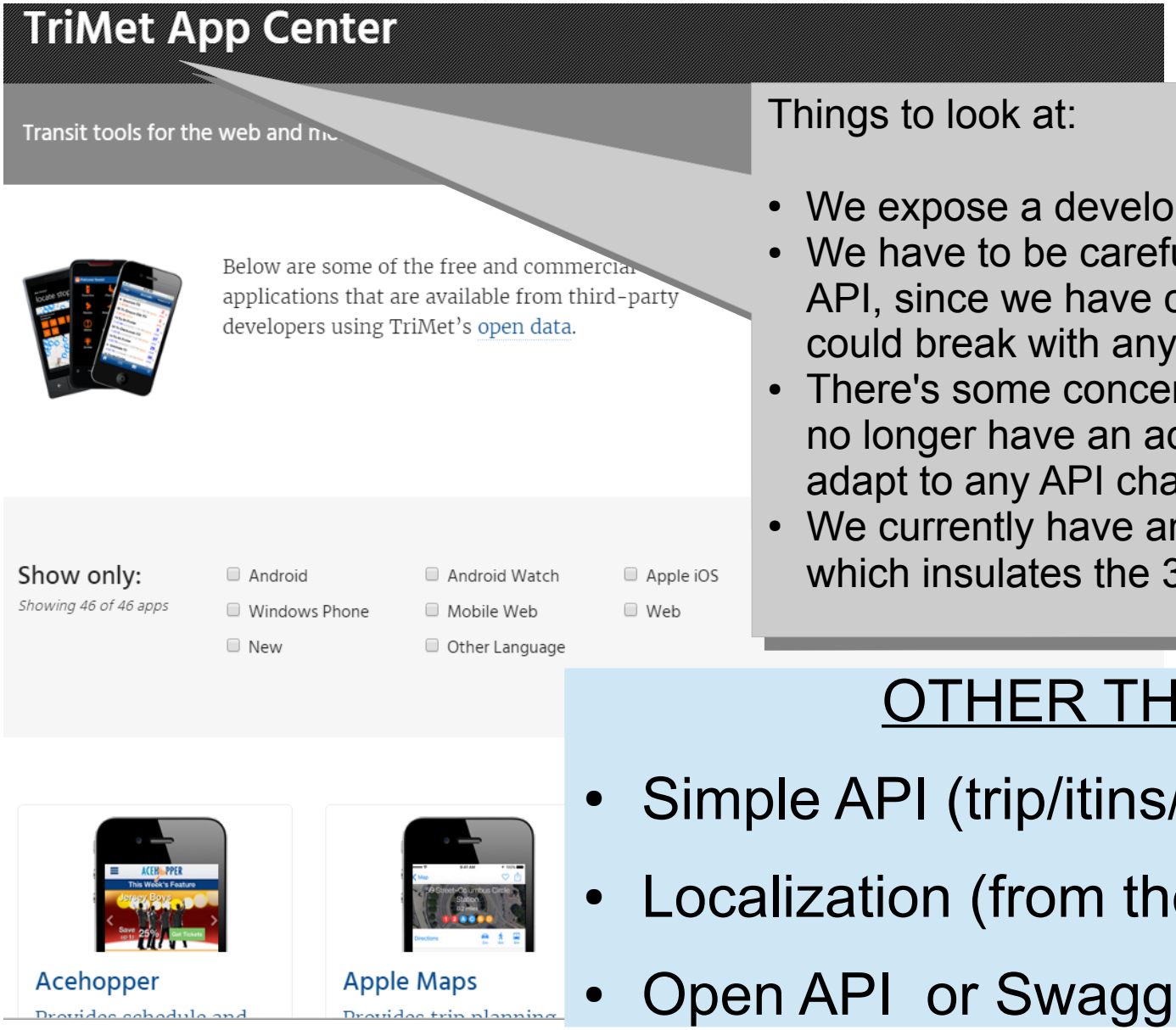
issue #5b: route stop / snow



Things to look at:

- Another 'amenity' for a stop is the ability to mark the stop as a 'snow' stop for a given route.
- How we would even represent this via GTFS is an unknown at this point.
- But it is something we've done in the past with data from TRANS (not gtfs).

Issue #6: developer.trimet.org



Things to look at:

- We expose a developer API to 3rd parties
- We have to be careful with changes to this API, since we have customers using apps that could break with any API change
- There's some concern that existing apps may no longer have an active developer who can adapt to any API changes.
- We currently have an API *Wrapper* atop OTP, which insulates the 3rd party apps from change

OTHER THOUGHTS

- Simple API (trip/itins/legs = complex)
- Localization (from the API)?
- Open API or Swagger.io api docs

Pros & Cons to using GTFSdb

(rather than OTP transit index)

PROS

- GTFSdb is successfully running at TriMet now
- If OTP React is built to be back-end agnostic == more flexible & adaptable system
- No need to *fix* the OTP index to work with TM data
- OTP should focus on being the best *routing engine* it can be..original architecture
- *Wrapper* insulates 3rd party developers from OTP change

CONS

- Requires development here at TriMet (OSS)
- Adds complexity to OTP React front-end code
- Might want to fix OTP regardless of this project
- It takes more effort to host, maintain and populate more than one system
- *Wrapper WS* is not (yet) widely adopted by transit app developers