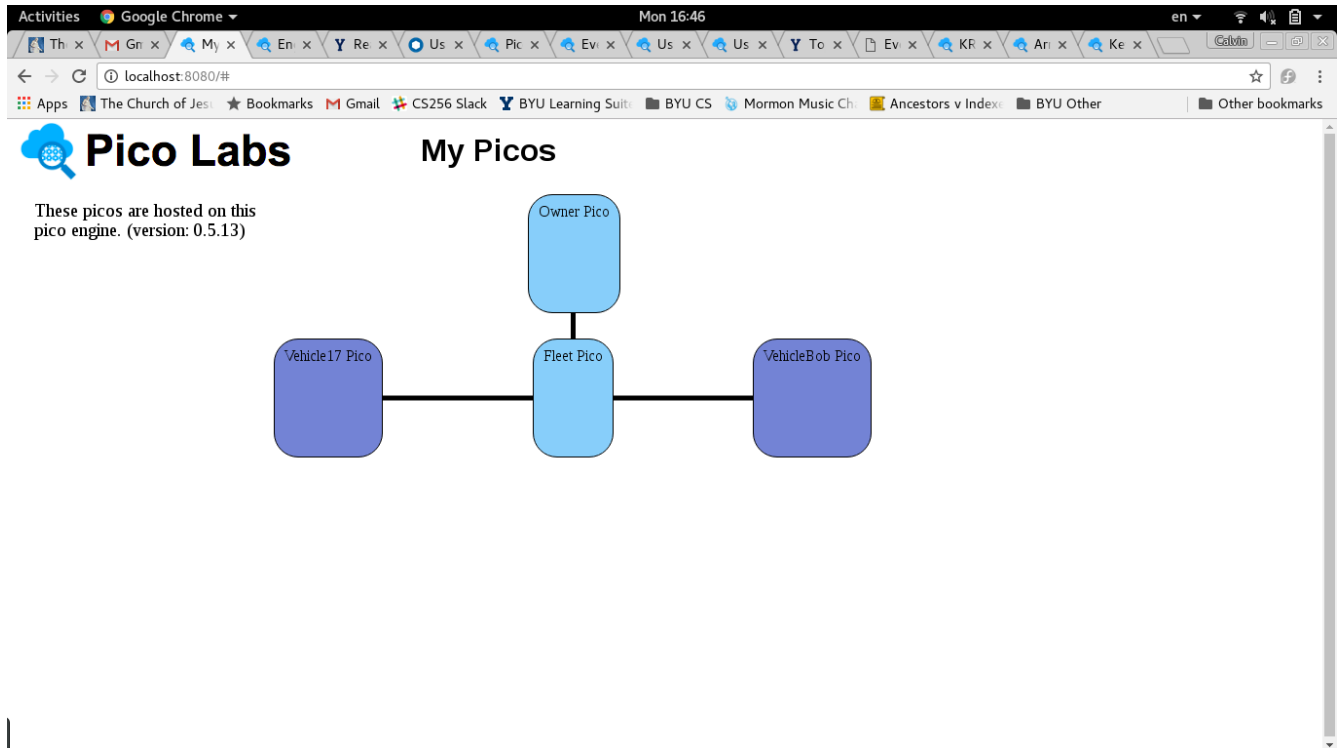


Deliverables

1.A diagram showing the relationships (i.e. subscriptions) between the picos.



2.The source URLs for your rulesets

https://github.com/calvinMcM/462_KRL/tree/master/Multi/partI

3.The RIDs for your rulesets

manage_fleet

4.ECIs for the picos that your rulesets are installed in.

cj09m68vi0001xtqw1e0ejvd6

cj0x0e7hf0001lbqwtgl9z3d1

5.The ECIs and events for adding trips to the vehicle picos.

cj0ycjx7a0009lzwgeal3vxs

cj0ybp7n80005lzwqweajfcs8z

6.A URL for generating a fleet report via the function method.

[http://localhost:8080/sky/cloud/\[ECI\]/trip_store/trips](http://localhost:8080/sky/cloud/[ECI]/trip_store/trips)

7.An ECI and event for generating a fleet report via the scatter-gather method

`cj0x0e7hf0001lbqwtgl9z3d1`

`fleet:gather_trip_reports`

8.Answer the following questions:

Questions

1.Can a vehicle be in more than one fleet based on the code your wrote? Why or why not? What are the implications of a vehicle being in more than one fleet?

My code only involves one fleet, but could be adapted fairly painlessly for more than one fleet, since the only challenge is keeping rcn numbers unique between fleets. If the vehicle is in more than one fleet, it must do so through subscription, not by hierarchical bonds.

2.How did you solve the problem of only notifying vehicles that you needed a report?

I filter it with built in filter capabilities.

3.How could you ensure that only certain picos can raise an event that causes a fleet report to be generated?

You can't. Pico's open API means that I spam requests for reports from anywhere. Through subscriptions you could probably check to ensure that the sender does in fact have a suitable relationship to be able to call for a report, but that's about it. The trick is identifying a valid ECI from which requests are accepted in the selection statements.

4.How do the debug logs differ for Part 2 (Generating Reports via Functions) and Part 3 (Generating Reports via Scatter-Gather)?

A lot more calls and events are raised throughout the process.

5.How can you know a report is done and all the vehicles that are going to respond have reported?

When each vehicle reports, keep track of the number of reports. When it reaches the number of targets you sent requests out to, you've reached the end.

6.Given your answer in (3), how would you recover if the number of responding vehicles is less than the total number of vehicles?

Send the request out to the missing targets.