

IOT BZH

Vehicle 2 Cloud Signaling and Data collection



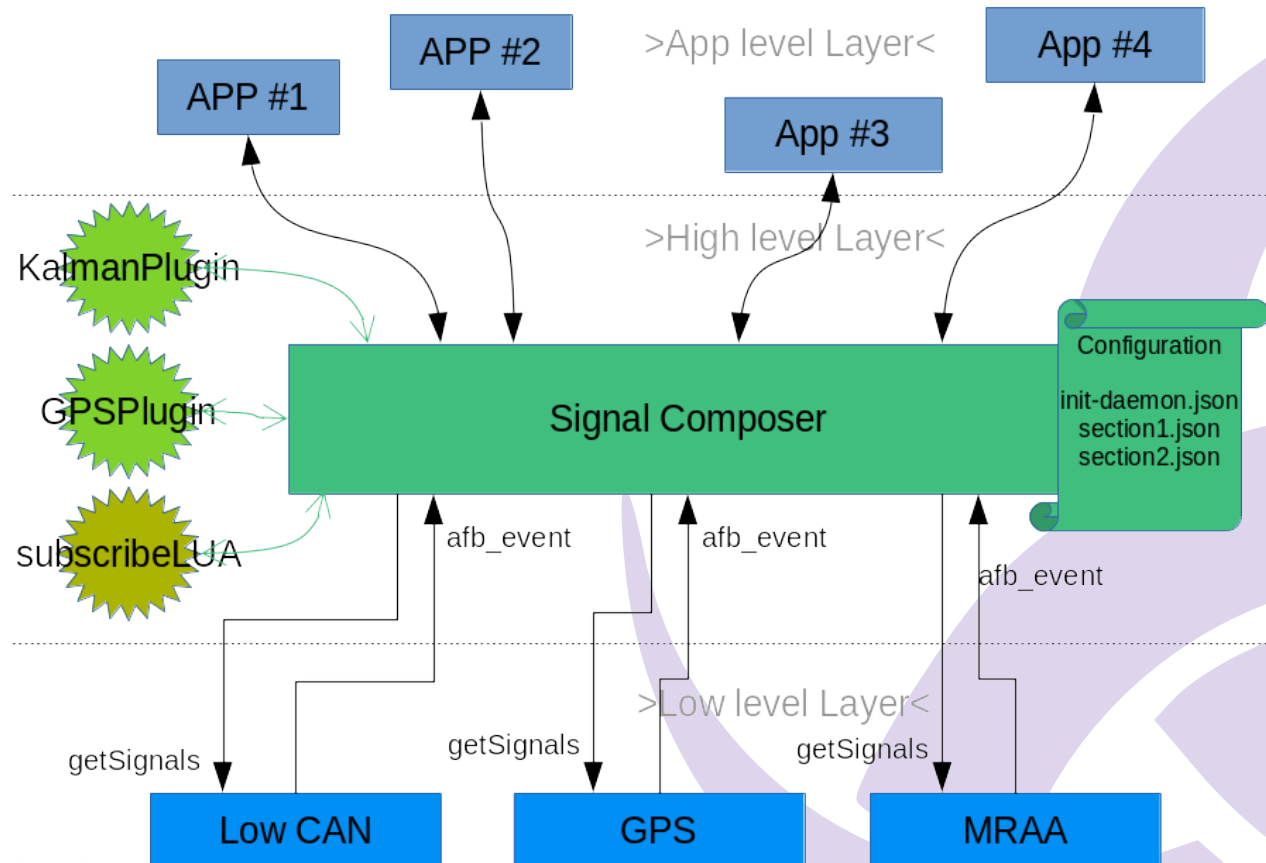
AGL Technical F2F@Microchip
April/2018

Romain Forlot & Sébastien Douheret



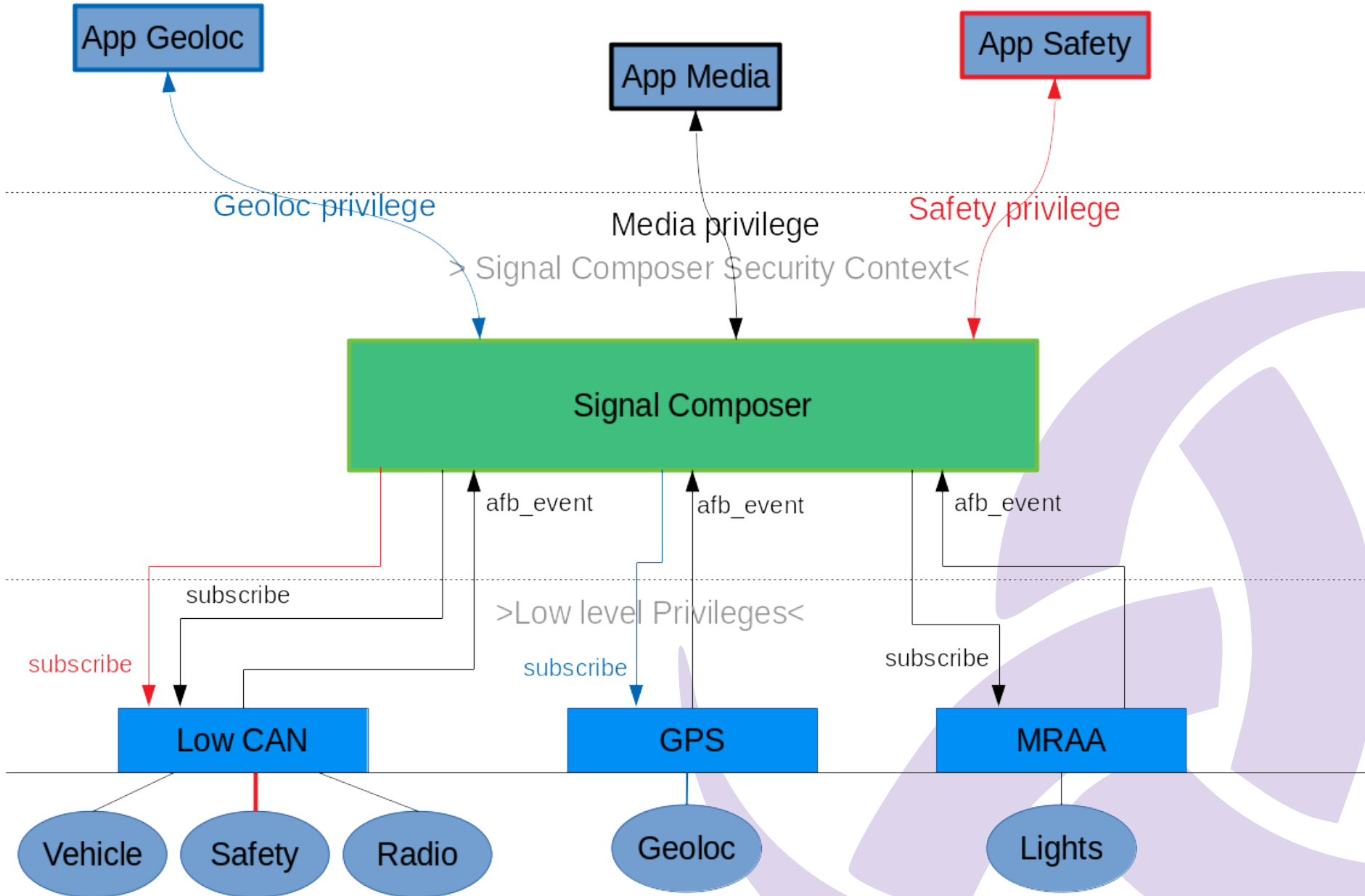
Signal Composer

- Updates:
 - Loading new sources and signals at runtime [done]
 - Tagging signals [To Be Done]
 - Secure access to signals (privilege) [To Be Done]
 - CAN Gateway configuration to access signals [To Be Done]



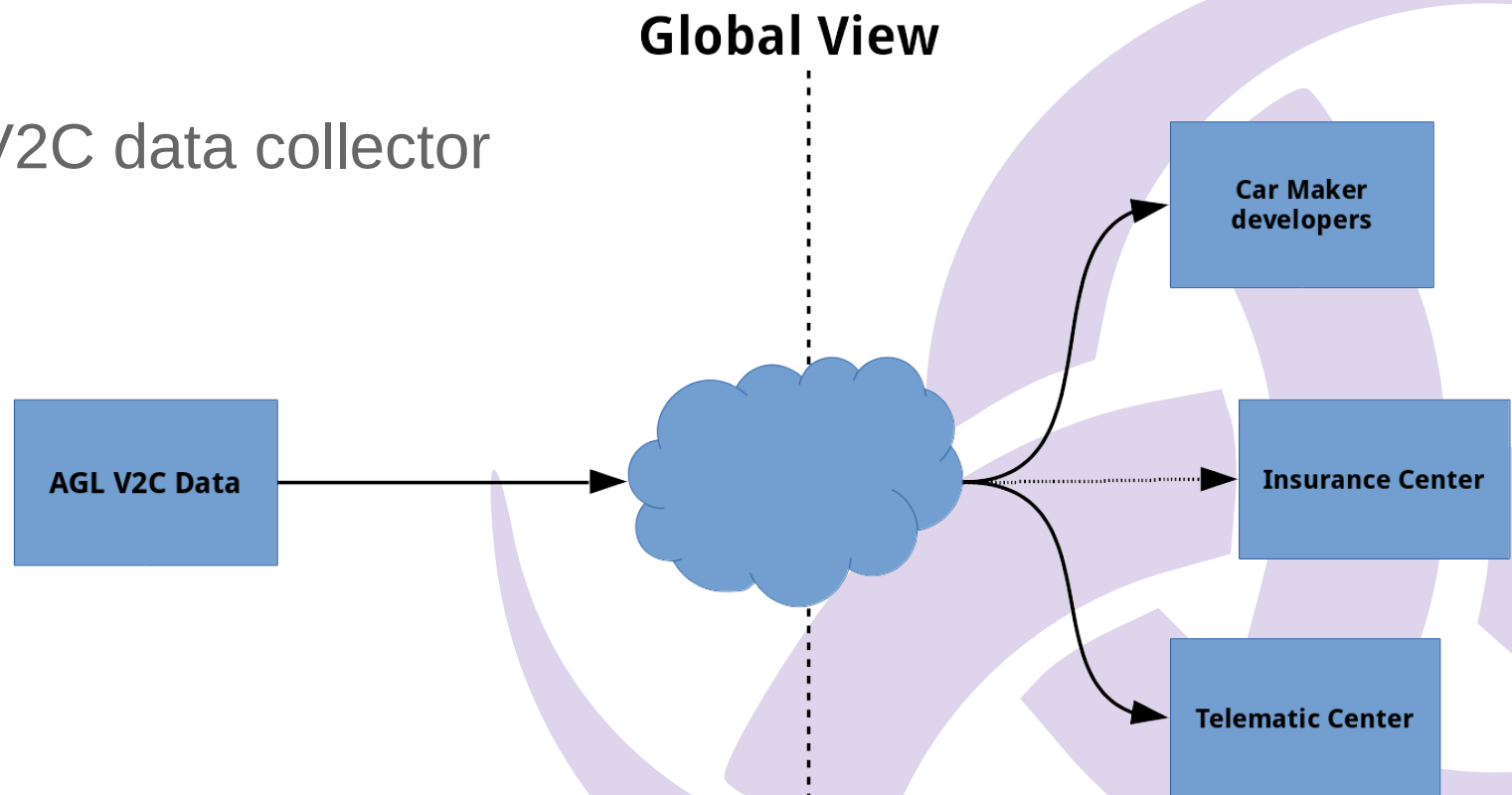
Signal Composer

>Client Security Contexts<



Vehicle → cloud

- 2 needs :
 - Telematics
 - Debugging data and tuning
- 1 way :
 - AGL V2C data collector

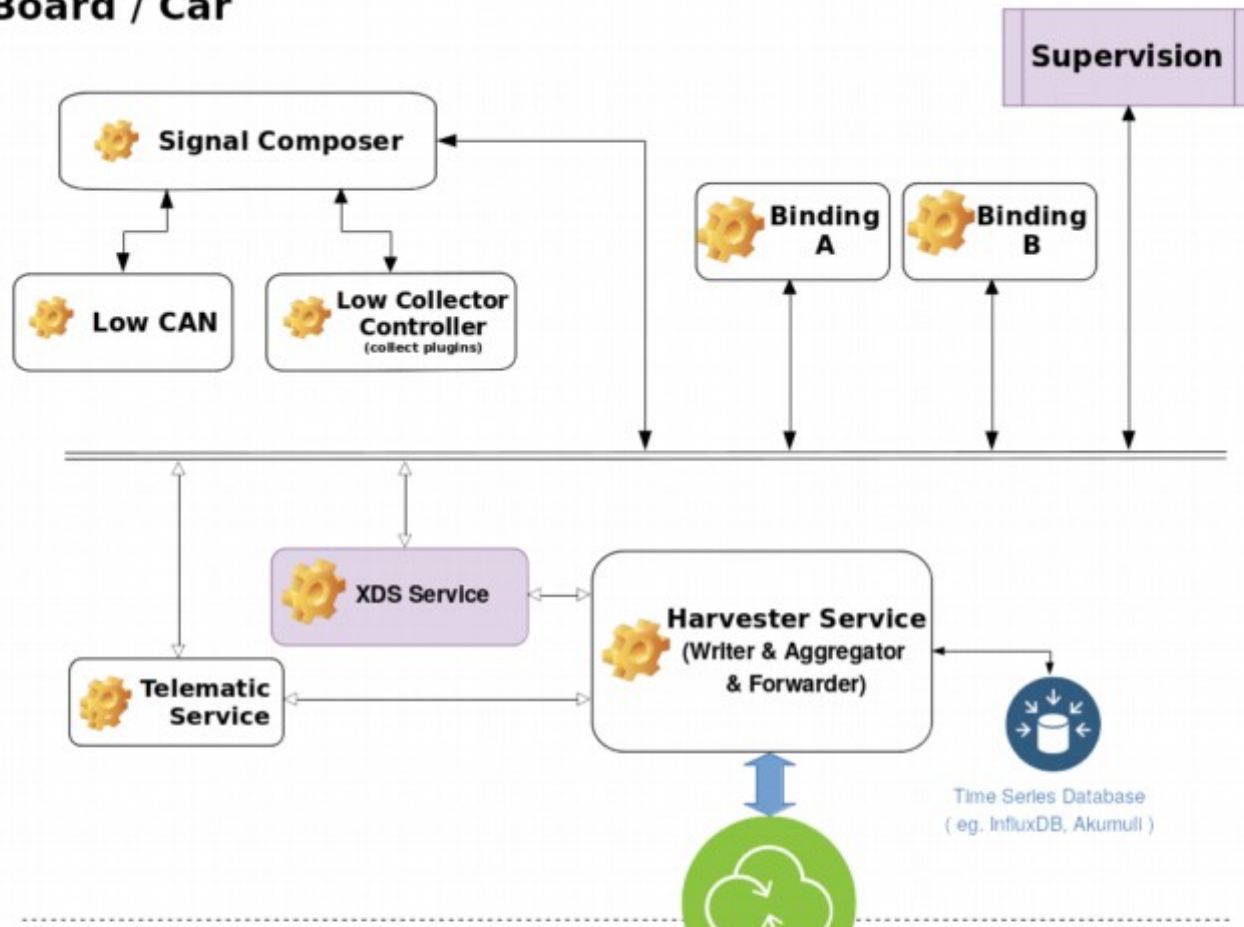


Vehicle → cloud

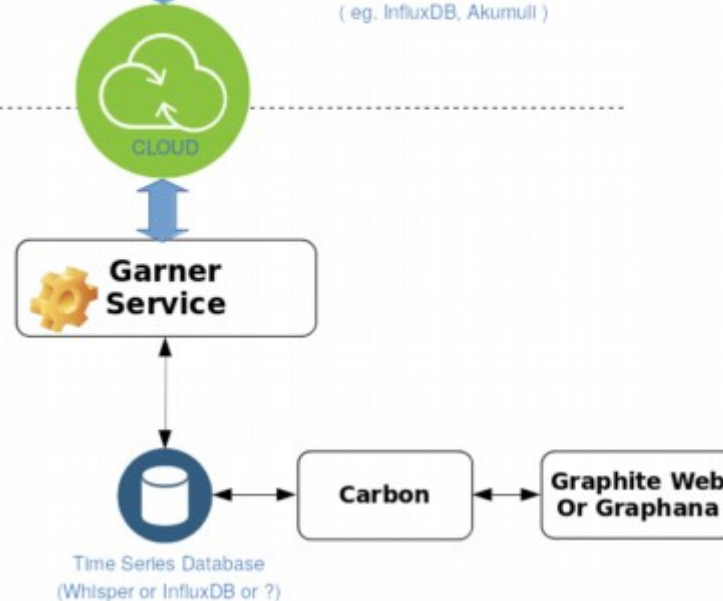
- Telematics
 - Based on acquisition CAN, vehicle signals, GPS, ...
 - Externalized vehicle data analysis (failure detection)
 - Black box, Save box
 - Push useful data to the cloud (aggregation, retention policies)
 - Development and Production mode
- Data collection for debugging
 - Collect system data (cpu, memory usage, ...) and AGL binding exchanges (supervision / monitoring)
 - Only in development mode
- Tuning
 - Car testing campaign (offline storage)
- Store caching: data saved on-board or in a cloud Time-Series database

Global Architecture

Board / Car

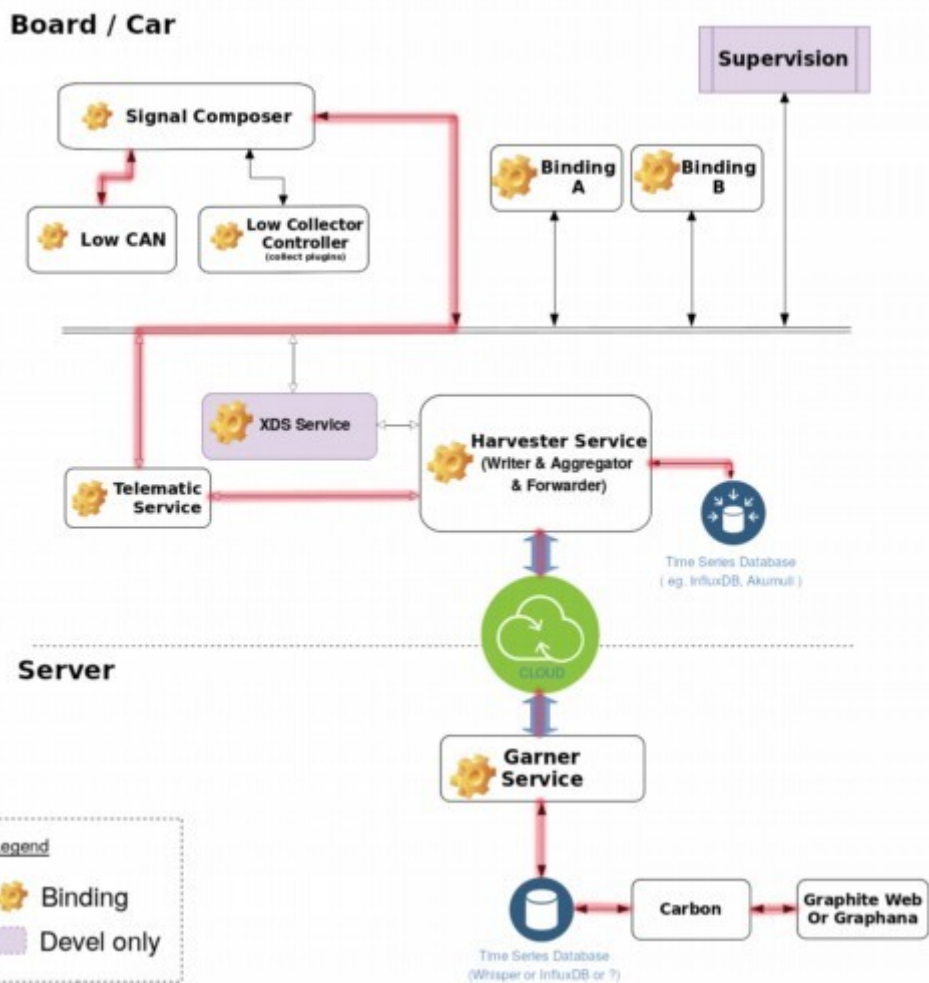


Server

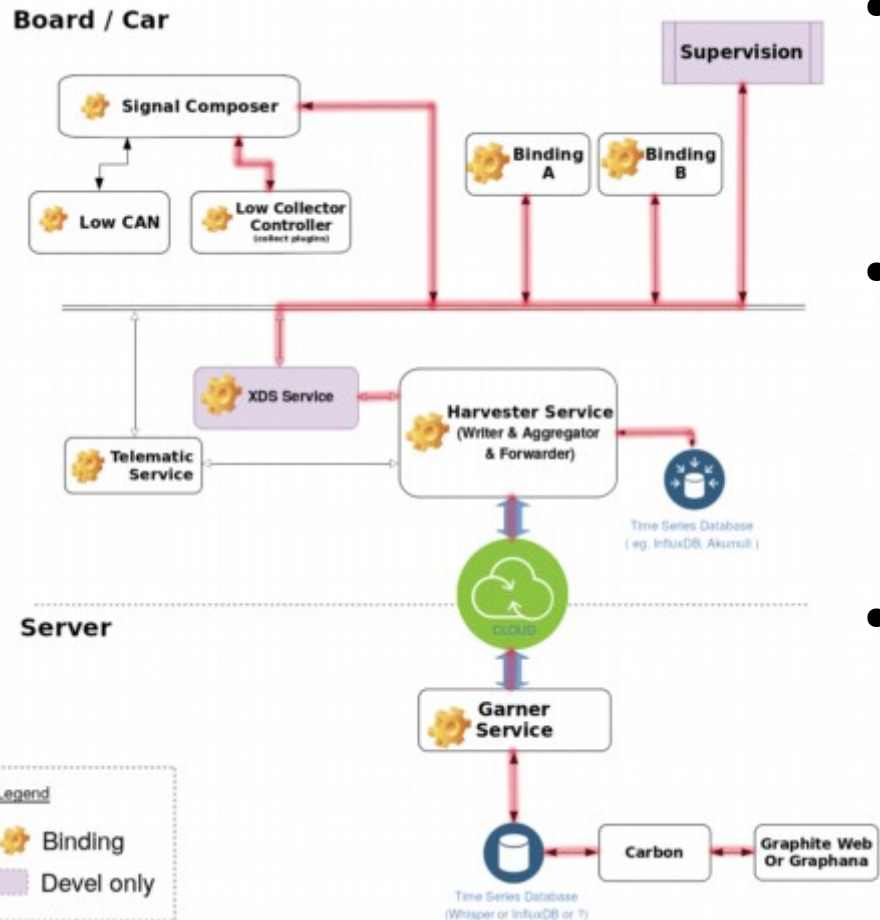


Telematic workflows

- Telematic service used to pilot Signal Composer
- Push signals configuration creating the ones not already existing
- Able to access specific protected signal on behalf higher permission from telematic privilege client.

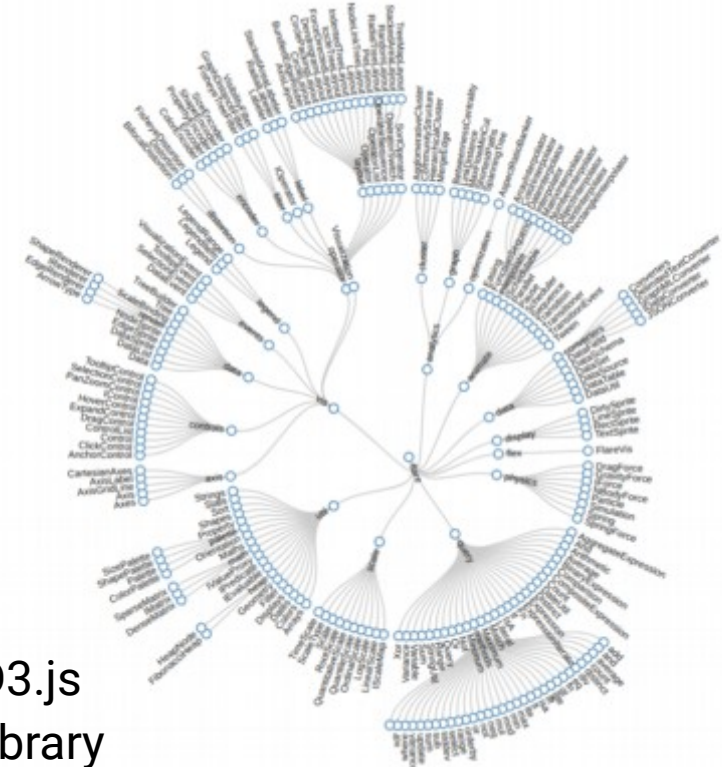
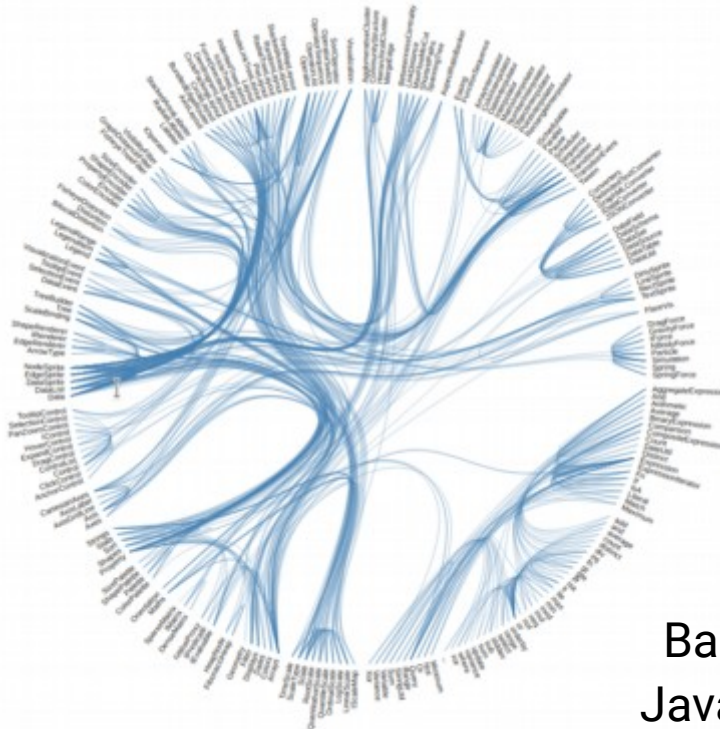
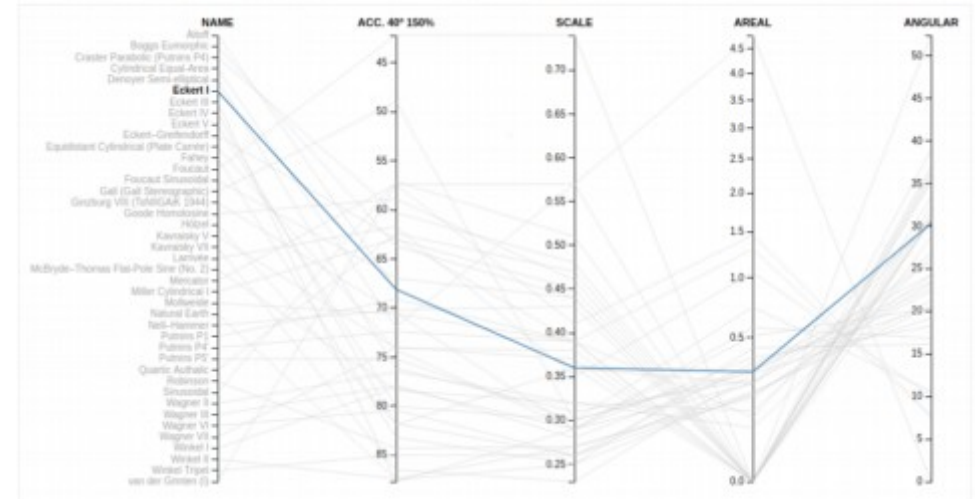
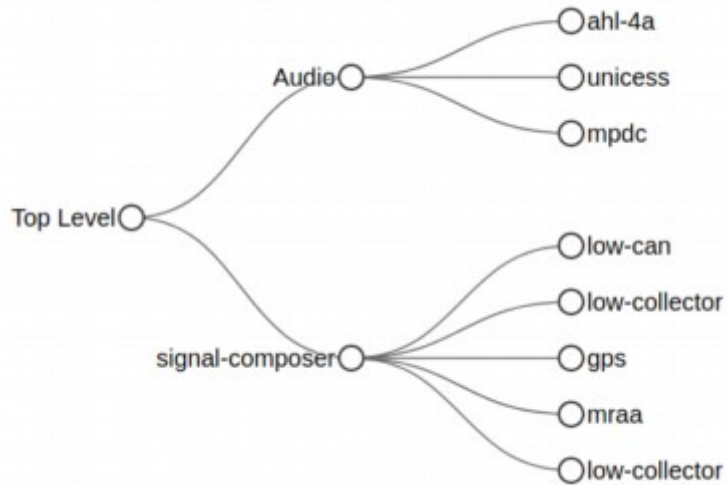


Debugging / Tuning workflows



- XDS Service binding used to configure which data to collect
- XDS relying on Supervision to collect bindings info and communications
- XDS relying on Signal Composer / Low Collector to collect system metrics (cpu, memory usage, ...)

Rendering examples to debug bindings exchanges



Based on D3.js
Javascript library

Technical challenges

- Local processing in Car / on board (NAND writes)
- Time Serie Database not initially designed for embedded world
- Collecting too many data = intrusiveness
- Tag data (ownership, priority, lifetime)
- Security privileges (request new feature of API V3)

Further Information

- Proof of concept sources
 - <https://github.com/iotbzh/agl-service-harvester>
 - <https://github.com/iotbzh/agl-service-xds>
 - <https://github.com/iotbzh/agl-service-low-collector>
- Time Series Database / Tools
 - <https://www.influxdata.com/>
 - <http://graphite.readthedocs.io/en/latest/overview.html>
 - <http://opentsdb.net/>
 - <https://grafana.com/>
 - <https://d3js.org/>