Empowering digital services

VISS API towards Android Automotive OS

AMM April 2024 in Gothenburg
Speakers

Emil Dautovic

Ulf Björkengren

Peter Winzell

Renjith Rajagopal

Kristoffer Nilsson
Things to solve

- Enable access to signals from a vehicle - both Android, no need for slow recompile, and non-Android
- Easy to add new use case - e.g. insurance that need access to new data for driver behavior
The different pieces of the puzzle

- RemotiveLabs
  - CLOUD COLLABORATION
    Work collaboratively and share data with colleagues & partners.
- SOFTWARE DEVELOPER
  Wants to #getstuffdone
- Remotive Broker
- VEHICLE
  x 1000 signals available
- VISS
  Standard for accessing vehicle information
VISS reference implementation overview

- The Vehicle Information Service Specification development was started at W3C
  - [https://github.com/w3c/automotive](https://github.com/w3c/automotive)

- Where it was discontinued and in March 2024 migrated to
  - [https://github.com/COVESA/vehicle-information-service-specification](https://github.com/COVESA/vehicle-information-service-specification)

- VISSv2 features
  - Supported transport protocols: HTTP / Websocket / MQTT
  - Methods: Get / Set / Subscribe (Unsubscribe)
  - Subscription filters: Timebased / Change / Range / Paths / Curve logging / History
  - Access control: Token based, OAuth2 inspired, RBAC support, single node granularity
  - Consent support: Obtaining consent delegated to an External Consent Framework, acts as firewall
VISS reference implementation overview

- The VISS reference implementation (vissr) is after migration found at
  - https://github.com/COVESA/vissr

- vissr features
  - Complete VISSv2 feature support, plus
  - gRPC
  - SwCs to realize a complete tech stack:
    - Data store: SQLite / Redis / IotDB / ..
    - Feeder: Template where only the vehicle interface client needs implementation / RL-feeder / EVIC-feeder / ...
    - Client: JS based: AGT, AT, HTTP, WS, integrated access control,.. / Go-based: MQTT, curve log to CSV file, ...
    - Domain Mapping Tool: creates feeder conversion instructions from YAML input
Demo SW stack

AGT API

RL broker

RL feeder

Redis

VISS API + AT API

Application

HTTP
gRPC

Vehicle data recordings

18 April 2024 | Copyright ©2024 COVESA
Curve logging

- A method of moving data efficiently from vehicle to server.

- The purpose of the algorithm is to identify useless points and discard them while preserving the most important information.

- It keeps checking the points of maximum error. If the difference between the predicted and actual position is significant, the points are sent. Throw away the points that, essentially, were not carrying any useful information.

- Method developed by GeoTab (based on the Ramer-Douglas-Peucker algorithm). Now part of the VISSv2 specification
Demo content

- Use case: “How could an OEM enable access to certain Vehicle datasets to digital partner that want to accurately calculate Safety Scores based on driving behavior”
  - Safety Score Android application/service - Runs on AAOS generic emulator
    Request AG token (HTTP) - Authentication/Authorisation
    Request AT token with AG token (HTTP) - Validate access
    Subscribe Multiple VSS data with AT token (gRPC)
  - VISS + AT server - Runs on local host PC as docker container
    VSS datasets tagged with ‘validate’ attribute with access-control mode - “read-write”
  - AGT server - Runs on Local host as docker container
    Authentication & Authorisation of client
  - RemotiveLabs virtual sensor cloud - Runs on cloud
    Pre-recorded real vehicle datasets collected from Volvo XC90 PHEV
Demo time
Conclusion

• An expansive, reputable & industry adopted Open Vehicle API ecosystem that respects Security, Privacy & Consent beckons new thinking & fosters innovations.
References

More interesting stuff at:

- Sign up for free and test at [https://cloud.remotivelabs.com/](https://cloud.remotivelabs.com/)
- The VISS specification at [VISS version 2 - Core (w3.org)](https://www.w3.org/2016/viss/core), [VISS version 2 - Transport (w3.org)](https://www.w3.org/2016/viss/transport)
- The VISS reference implementation at [https://github.com/COVESA/vissr](https://github.com/COVESA/vissr)
- Reference android application at [Android application](https://example.com/android)
- GeoTab curve logging at [curve logging](https://example.com/curve)
Thank you for your attention