20200514-Virtual-Technical-Summit-CCS-Workshop-Minutes

| 1 | ## Introduction |
|---|--|
| 2 | Philippe Robin welcomes everyone and hands over the mic to Kevin Valdek |
| | 50 attendees |
| | slide deck for this workshop is here |
| 3 | ## Project overview |
| 4 | Kevin: this is the presentation Gunnar and I delivered in the Automotive World webinar last week, recording is available |
| | Introduction slide #4: Project charter Even though cars are connected, they are limited to specific OEM/brands A lot of ideas and start ups introducing new services We want to harmonize these activities Define building blocks with flexibility for common solutions Define data exchange and access slide #5: Vehicle data |
| | possibly on object storage in a later stage Buffering (high frequency, poor connectivity) to take into account slide #22: Neutral servers and data marketplaces opportunity included (in accordance to ISO20078) or simple 3rd party access |
| 5 | discussion on framework architecture |
| | Iyyaz Baber: What about a noSQL DB in the cloud? Kevin: We could. In the end, the proof-of-concept came from our experience with relational DB, there is no restriction Ulf: the server exposes an API. As long as it follows the API, the DB could be of different types. Gunnar: this would depend on the data (time series fitting into a relational DB ?). Teddy Zhai: When you say: "Cloud is Client, and in-vehicle is Data Server", do you mean in a logical view or technological view? It means, for example, Data Server needs to open ports and listen on them. Ulf: it's both, the data servers needs to open ports and listen, and response to requests (standard client-server model) Gunnar: it can vary from this current proof-of-concept architecture to a production-ready solution. Typically there could be an SSL tunnel set up by OEMs Teddy: This is a major architecture decision in our view. It would be great if GENIVI can provide a threat model for it. Gunnar: I think in the later stages of proof-of-concept development, it is worthwhile to address these questions about how to set up private networks between some parts (e.g. OEM to car). If we end up using SSL/TLS then the GENIVI Security team has previously published a best protective private at of security private. |
| | François Fischer: data model - you can add SAREF/Autopilot Philippe: we had a look at SAREF proposal in September 2019 (SAREF Automotive - https://www.w3.org/2019/09/trans-data-ws/SAREF.pdf) and estimated that it was more on the long term (5+ years) goals, it might be worth rechecking the status of SAREF work Francois: I coordinated a project about IoT and automated driving, i.e. AUTOPILOT and we pushed data models to SENSORIS and SAREF, see https://autopilot-project.eu/ Cumpar: theore are detected included in the current VSS work, we expect to use an Electric Vohicle (EV) use case and data set for the proof of |
| | Guinial, there are datasets included in the current VSS work, we expec to use an Electric Venicle (EV) use case and data set for the proof-of-concept demo implementation Guiniar: our objectives are to get fast results, identify technologies and components that need to be standardized |
| 6 | ## proof-of-concept demos |

7 Demo #1 - OEM cloud

- Ulf: shows a Cloud and Connected services demo involving 3 components:
 - data server (W3C Gen2),
 - ° data base (Open Vehicle Dataset Server a.k.a. OVDS)
 - CCS vehicle client
- Data server can be accessed through multiple protocols (HTTP, WebSockets) with VSS(2) data (but open to use other compatible domains with a service manager)
- Data base is a SQL database with a VSS-to-DB adapter
- CCS client sits in between and creates a list of all VSS leaf nodes and issues read/write requests to the Gen2/OVDS servers

8 Demo #2 - Vehicle data generation

- slide deck is here
- Stefan Vysocki (from the GENIVI AASIG Vehicle Data / VHAL project) shows a demo of a vehicle data feeder using a vehicle simulator and a GraphQL server
 - simulator used in the demo is OpenDs, a frame example is given (could be VSS too)
 - ° the demonstration iincludes the simulator interface and a GraphQL server to retrieve values from the simulator
- Kevin Valdek: Could we use a fleet instead ?
- Stefan: Yes, by configuring multiple ports
- Stephen Lawrence: Could you containerize the simulator and automate tests ?
- Stefan: It is partially containerized, no issues.

9 Demo #3 - GraphQL client on top of a VSS schema

slide deck is here, Kevin shows a demo

Discussion

- · How to have data anonymization ?
- there is consent here
- What about the identification?
- Here we could have many types of identifiers, such as what a brand allows you to access (a certain fleet for instance)
- What happens when the user removes consent (need to do something on the 3rd party side)?
- These are great use cases, we should look into them.
- How do we categorize what is to be anonymized and what is not?
- Since privacy is partially region-based, it might be better to have configuration outside of the VSS

10 Discussion on technology options

- · Gunnar introduces the topic with this slide deck
- How to generate data
 - Usefulness of an easy to run driving simulator
 - it would be nice to use the simulator as a driving game and generate data
 - Iyyaz: I go for it !
 - Gunnar: which operating system would you use ?
 - Iyyaz: ubuntu, linux or windows 10
 - Christian: having vehicle data would be useful to test our APIs !
 - Stephen Lawrence points out that CAN can-utils has the basic cansend to send can msgs, look at https://github.com/linux-can/can-utils, and https://manpages.debian.org/stretch-backports/can-utils/cansend.1.en.html
 - How important is the sanity of the data ? Could we do statistical analysis on random data? What about randomly generating data ?
 - Gunnar: Is that sending a random generated data you mean?
 - Stephen: can-utils also has cangen for random data
 - Stephen: it seems also that candevstudio may have some generation capability, look at CANdevStudio repository
- Alternative technologies
 - Why not using MQTT between the vehicle and the cloud (rather than other than W3C Gen 2) ? with MQTT there is a need to define the payloads
 - using maybe a NoSQL data base
 - Abhijeet: mentions the use of a replica of the car like AWS Car Shadow and like the Ditto digital twin Device as a service: Ditto project from Eclipse https://www.eclipse.org/ditto/
- How to do high-volume tests ?
- Evaluating the performances ?
- · What about security ?