Adding gRPC Support to the VISSv2 Reference Implementation

By Ulf Björkengren, Senior Software Architect, Ford

The Vehicle Information Service Specification version 2(VISSv2) developed in the W3C is a service for accessing vehicle data, reading from sensors, or writing to actuators on the vehicle's network. It exposes this data using a hierarchical tree-like taxonomy defined in COVESA Vehicle Signal Specification (VSS). The service provides the data in JSON format. The service may reside in the vehicle, or on servers on the internet with information already brought off the vehicle.

VISSv2 specifies three transport protocols that are allowed to be used in client-to-server communication. The Websocket and MQTT alternatives are payload compatible, so a client can issue requests over either of these transport protocols without modification of the payload. HTTP where the path information is explicit in the URL is not fully payload compatible. In a typical scenario, the server is deployed on a high-performance ECU running Linux, and clients can either be deployed off-vehicle (e.g., in the cloud or on a mobile device in proximity of the vehicle), or in-vehicle then typically on another HPC ECU, or the IVI ECU.

As described above, the VISSv2 specification supports a wide array of scenarios but there are of course scenarios not covered. To fill a bit of this gap, an experimental addition of gRPC support besides the other transport protocol alternatives can now be found on the VISSv2 reference implementation. The implementation is payload compatible with the Websocket and MQTT alternatives, and it can be configured to run over TLS. Using protobuf for the serialization, the payload is after protobuf encoding in binary format and obviously not identical on byte level to the others where the payloads have text format. But after protobuf decoding on the receiver side, the payload is again identical also on the byte level. So, a client making use of the protobuf transcoding interface can use the same payloads on any of these transport protocols.

The reference implementation contains a client test implementation written in Golang that uses this transcoding.

Adding gRPC to the suite of transport protocols that VISSv2 supports gives a client implementer a wider choice in finding a protocol that meets the needs of the client use case.

Whether this transport protocol will be added to the coming W3C standard is at this point not decided, but if you believe it should after testing it on the reference implementation, you can mail your support for it to Paul Boyes, COVESA's Community Director.