Data Access Control

This page intends to document:

1. Existing standards and specifications for access control mechanism to vehicle data
2. Example of definition of access groups / permissions / roles

NOTE: The page is not intended as a comprehensive analysis of different choices*. For 1) it documents only those access control mechanisms that may be applicable for *currently investigated technologies* in CCS. Similarly, for 2) we expect to have examples of applying those principles to the *currently investigated data model*, which is now assumed to be VSS. When the range of technologies that investigated in CCS increases, then the access control mechanisms provided by those technologies mightalso be analyzed.

* As a counter-example, general research into *all* available strategies for access control (Role-based systems, typical application permission systems such as those for Android apps, and many other examples), is hugely important but kept out of scope for this page. Such research of already existing knowledge ought to be done whenever a new mechanism is defined, but we expect that to be collected elsewhere, linked to each such concrete project.

As an example, existing standards that inspired the access control definition chapter in the W3C VISSv2 protocol were collected through discussions in the W3C Automotive working group and documented there in meeting minutes and in discussions on mailing lists.

Access control mechanisms related to CCS activities

W3C VISS v2:

- For W3C VISSv2 access control, please see this chapter in VISS v2.

Other technologies somewhat investigated in CCS project:

MQTT:

- **Client Authentication**. When a connection is established the client must authenticate before a connection is established. A simple method is user/password but it can use an extended authentication described in MQTT 5, including multiple challenge-response message exchanges between client and server. It is generally assumed that the client-server exchange will be protected by TLS as a starting point. The authenticity of the provided TLS certificate is expected to be the main method for a client to ensure the server is authentic.
  - In the client connection with extended authentication, any method can be named as a string by the client, and the server responds only if this one is supported. Using method names (and procedures) described by SASL seem to be recommended, but beyond that there are no requirements in the MQTT specification. The particular details must be agreed between server and client and implemented with compatibility on both sides. In other words, when reusing implementations, it is necessary to checked what they might support regarding authentication.
- **Topic access control**: A server could theoretically/optionally limit subscription to particular topics based on the identity (or theoretically any other credentials exchanged in the authentication sequence) that was associated with the active connection when the initial authentication was performed. Limiting parts of the topic tree to certain clients seems also to be not described in detail. In other words this must also be implemented in some agreed-upon way which is left out of scope of the MQTT protocol specification itself.
- **Conclusion:** There seems to be work to do here to define, and implement, the mechanism specifically for VSS signal access, based on the topic tree defined from VSS.

- Options in Apache NiFi / related technologies? - **TBD**
- Access control principles defined by WAMP – **TBD**

Example of defining roles/permissions to VSS signals

- Example of definitions of role/purpose groups applied to VSS, according to W3C VISSv2 concept. – **TODO**