

CVII-TS: VISS v2 server implementation in C++

This is a planning / work breakdown page for driving the development of a second full-featured (?) VISS v2 server (in addition to the comprehensive [GoLang version](#))

Design constraints / requirements

- Group rough consensus seems to be that C++ is the preferred language.

Background information

Driving forces behind this

- Go code compilation in embedded still not fully supported everywhere
- W3C has generally a requirement to have two full implementations "proving" the specification before it can become fully official.

There are at least two known starting points that are C++ based, but these currently support a limited VISS v1 feature set.

- [KUKSA.VAL](#)
- Implementation used in AOS [aos_vis](#) written in Go (a related code is the [libvis](#) but that is a *client* implementation in Python)
- KUKSA.VAL (is basically a VISS server, but some other protocols are being added). Gradually started work on increasing the scope to v2.
 - Any Linux-specifics in implementation?
 - Implemented a connection to SocketCAN Linux specific API, (but of course other input modules could be added)
 - Not sure about details but development and testing is on Linux.
- Both KUKSA.VAL and AOS versions added some access control, but not according to v2 spec.

Access Control

VISS v2 has a significant update of the access control description. If this is going to be a *full* implementation then it should implement access control according to specification.

Other Features

The v2 specification has a lot of flexibility in queries, various types of filters, and service-discovery like information such as available signals. There is an ongoing discussion of whether some features should be marked as optional in the specification but it has not been completed yet.

Naturally, any implementation should track the implemented features in some kind of matrix.