VISSv3 Feedback 3

Miscellaneous

There was the desire to have a standardized way to get notified of all changes. (even for non-int datatypes).
We discussed this behavior for a subscribe without any filters but this was seen as a bad default behavior.
Nonetheless as far as I remember it there was general agreement that this might be useful for some situations.
-> consider adding these semantics explicitly for timebased = 0

In the JSON Schema it looks to me like the *number* inside *error* is not required.
This seems to contradict (5.2.1 Error Information) link.
Also the exclusive enum for the *description* is more restrictive than what Transport defines.

The "*Server Tree Example* link is not an example. Because: "Below follows an example of a server capabilities tree.
The tree *MUST* contain the structures listed below[...]" This seems to clash. Either it is an example or it is mandatory.

Errors

> Error Feedback
>
> ## Current status in AAG
>
> - Many errors are re-defined in AAG specification.
> - They mostly use the sentence pattern (highlighted parts are variable)
>   - The Automotive API Gateway shall respond with an Error Response to a VISS ... request in case ....
>   The Error Response in this case shall have the Error Reason ..., the Error Code ..., and the Error Message "...".]
>   - For errors that are not (yet) part of VISS
>   - [...]The Error Response in this case shall have the Error Reason ... and the associated Error Code and Message according to [3].
>   - For errors that are mentioned in VISS (but not connected to this condition)
>
> ## Current status in VISS
>
> ### In Transport
>
> - Contains a list of possible errors
>   https://raw.githack.com/COVESA/vehicle-information-service-specification/main/spec/VISSv3.0_Transport.html#status-codes

§ 4.1 Status Codes

A server implementing this specification *SHALL* support the error codes, error reasons and error descriptions shown in the table below, for all supported transport protocols. The server may choose to dynamically replace the error description as described in the sub-chapters
The client *MAY* support any status code defined in [RFC2616].

Error Number (Code)	Error Reason	Error Description
400 (Bad Request)	bad_request	The request is malformed
400 (Bad Request)	invalid_data	Data in the request is invalid
401 (Unauthorized)	invalid_token	Access token is invalid
403 (Forbidden)	forbidden_request	The server refuses to carry out the request
404 (Not Found)	unavailable_data	The requested data was not found
408 (Request Timeout)	request_timeout	Subscribe duration limit exceeded
429 (Too Many Requests)	too_many_requests	Rate-limiting due to too many requests
502 (Bad Gateway)	bad_gateway	The upsteam server response was invalid
503 (Service Unavailable)	service_unavailable	The server is temporarily unable to handle the request
504 (Gateway Timeout)	gateway_timeout	The upsteam server took too long to respond

- Implementations can then add more specific Error descriptions for some specific error codes and reasons
  - A list of examples is provided
  - For errors that do not have such a subchapter the description can not be adapted

    § **4.1.1 400 Bad Request Error Descriptions**

    This error code and reason shall be used for JSON schema related errors. The default error description is shown in the table above. The server may dynamically replace this by any of the error descriptions in the list below, or by any other relevant error description.

    - Missing or invalid action
    - Missing or invalid path
    - Missing or invalid filter
    - Missing or invalid value
- Does not give any formal indication of when which error shall be used
  - From descriptions etc. some uses can be deduced but there is room for interpretation

In Core

- partially specific

  For representation of multiple data points, see 7.8.2 Response syntax.
  If data is represented incorrectly, then an error message with number 400, and reason "Bad data" *MUST* be returned. A specific case of this is if an array of data elements does not contain the number expected by the server. The server *MAY* then respond successfully, and follow a proprietary recovery policy, or it *MAY* respond with error number 400, and reason "Invalid array size".

  link
- partially unspecific

  § **5.1.2 Update**

  Purpose: Provide an altered value to the vehicle signal addressed by the path.

  The client *MAY* have to obtain an authorization token before being able to update the vehicle signal. If the server is able to satisfy the request it *MUST* return a success response, else it *MUST* return an error message. Only actuator type signals can be updated. Please note that a success response does not guarantee that the actuation attempt to change to the updated target value has, or will, succeed. A client may monitor the actuation progress through subsequent reads of the actuator value.

  link
- mix

  § **7.8.1 Error Handling**

  A request addressing multiple nodes may address both valid nodes, and invalid nodes. The latter case shall lead to a Forbidden error (403) response message part that contains information about which node, or nodes, that are invalid. The error message shall not contain data from any of the validly addressed nodes.

  link
- Some error situations are not explicitly mentioned in VISS
  - E.g.:
    - invalid upstream server response
    - Unsupported feature requested

Consequence

- **Clients can not be sure what went wrong when they receive a specific error response**
- **Implementors can not be sure when they are supposed to send a specific error response**
  - Have to guess based on common semantics and error description

- Protocols that do not support the HTTP error codes can not support parts of core
  - This applies for the *specific* and *mix* situations shown above
  - Not critical since all defined transports support these codes (albeit with small application layer)

Suggestions

- Define which error to return in which situation more formally
  - A suggestion of possible connections is presented in the tables later in this pdf.
  - Basically: We have defined this connection in AUTOSAR already and would like to contribute it to VISS where we think it can be generally applied
- Cover some more error situations explicitly
- Allow the adaptation of *Error Description* for all errors
- Remove duplicate specification from AAG
- Optionally (Not required for the current transports as far as I could tell)
  - Support some indirection to also allow different errors depending on the transport
    - Core references a specific error situation ID and transport then defines the *number, reason, description* for a given set of transports and that ID

Allow the adaptation of *Error Description* for all errors

§ 4.1 Status Codes

> A server implementing this specification *SHALL* support the error codes, error reasons and error descriptions shown in the table below, for all supported transport protocols. The server may choose to dynamically replace the error description as described in the sub-chapters
> The client *MAY* support any status code defined in [RFC2616].

link
Change to something like:
The server may choose to dynamically replace the error description ~~as described in the sub-chapters~~ by any of the error descriptions in sub-chapters below, or by any other relevant error description.
Background:

- The sub-chapters only describe replacing the description for some but not all errors
  Update the JSON schema to allow flexible descriptions:

```
            ]
        },
        "description": {
            "description": "The access token",
            "type": "string",
            "enum": [
                "The request is malformed.",
                "Data present in the request is invalid.",
                "Access token has expired.",
                "Access token is invalid.",
                "Access token is missing.",
                "The server refuses to carry out the request.",
                "The requested data was not found.",
                "The server is temporarily unable to handle the request."
            ]
        }
```

link

Define which error to return in which situation more formally

- Suggestion for a mapping between error condition and error response
  - Additionally: information of how the condition is currently represented in VISS
- While looking at these I also found some additional bugs etc. that I tried to highlight were possible e.g. "in conflict with core"

Nr	Condition	Error	VISS Transport representation	AR Spec	VISS Core Condition(s)	AAPI and VISS are compatible
1	VISS Update request: in case the given value violates VSS datatype restrictions	400 - invalid_data - Data in the request is invalid	yes in conflict with core (the error core specifies is not part of transport)	AAG_00102	Not very explicit in Core "If data is represented incorrectly...." link Mandates code 400 and reason "Bad data" which is not part of transport.	no
2	VISS Update request: in case the given value violates VSS datatype restrictions - especially array size	400 - invalid_data - Data in the request is invalid	yes in conflict with core (the error core specifies is not part of transport)	AAG_00102	"if an array of data elements does not contain the number expected by the server....." link May be implementation defined allowed. Else: Suggests code 400 and reason "Invalid array size" which is not part of transport.	no
3	Signal is currently not available / Upstream provider is currently not offering that signal	404 -unavailable_data - Data temporarily unaccessible	Yes, as extension	AAG_03403	Could not find a specific location for this. The closest I found: "If the server is unable to fulfil the request," read subscribe "If the server is able to satisfy the request [...] else it MUST return an error message" update	yes

Nr	Condition	Error	VISS Transport representation	AR Spec	VISS Core Condition(s)	AAPI and VISS are compatible
4	At least one of the requested VSS nodes is generally unknown to the server (not part of the supported VSS catalog)	404 -unavailable_data - Data is unknown	Yes, as extension	AAG_20014	Could not find a specific location for this. The closest I found: "If the server is unable to fulfil the request," read subscribe "If the server is able to satisfy the request [...] else it MUST return an error message"	yes
5	Misused filter: E.g. time-based filter without or with invalid time period; change filter with invalid logical operators or missing diff field; diff filed with invalid value	400 - bad_request - Missing or invalid filter	Yes, as extension	AAG_20023 AAG_20018 (all miss invalid value)	"The primary payloads that are sent over any transport protocol *SHALL* conform with the JSON schema in this appendix" link But does not define what happens if that is not the case. JSON schema also still allows for invalid values for *time period* (e.g. -1) *diff* (e.g. "invalid")	yes
6	Incorrect filter for request e.g time-based filter on Read	400 - bad_request - Missing or invalid filter	Yes, as extension	AAG_20017	"The variants timebased, range, change, and curvelog are only applicable for subscription requests." link But it is not describe what shall happen if this is not the case. JSON schema allows this.	yes
7	Unsubscribe request with unknown *subscriptionId*	404 - unavailable_data - The requested data was not found	No, what is used at the moment does not fit very well also forbidden by JSON schema	AAG_20013	"If the server is unable to fulfil the request, then the server *MUST* return an error message." link -> very unspecific	yes but chosen error is not ideal
8	Subscribe: subscriber does not have access rights for the VSS path	406 - insufficient_priviledges - The priviledges represented by the access token are not sufficient.	no mandated in core but not part of transport - 401 might fit better? also forbidden by JSON schema	AAG_20011 (does not differentiate sufficiently from 404, also "not available for subscriber is not clear")	"If the server is unable to fulfil the request, then the server *MUST* return an error message." link "If any of the mentioned validations fail, the server *MUST* reject the access request." link core mandates an error that is not part of transport	no - AAPI defines different error

Nr	Condition	Error	VISS Transport representation	AR Spec	VISS Core Condition(s)	AAPI and VISS are compatible
9	subscribe with invalid filter (unclear what the exact condition here is - see Nr. 5 & 6 they should cover this)			AAG_20015 (imprecise rather remove)		
10	subscription without filter	400 - bad_request - Missing or invalid filter	yes, as extension	AAG_20010	"The primary payloads that are sent over any transport protocol *SHALL* conform with the JSON schema in this appendix" link But does not define what happens if that is not the case.	no, AAPI uses error that is not in transport

Errors connected to upstream server or specific AAG implementation

- I could not find anything about this in the VISS core spec
- I would like to suggest to mention in core that an upstream error can happen and to define that in this case the server shall return either the same error as the upstream server in case it also is a VISS error otherwise an error with code 502. (or something similar)
- Define error for the use of an unsupported feature
  - Options: 400 (easier to implement) or 501 (better feedback)
    - Since the client is supposed to use the "server capabilities" 400 seems more appropriate.

Nr	Condition	Error	VISS Transport representation	AR Spec
2.1	**In VISS** Upstream server response was invalid **In AAGW** (more specific) - during the required interaction with the Service Interfaces due to a VISS request, the Automotive API Gateway receives a ComErrc::kNetworkBindingFailure (AAG_03450, AAG_03451) - during get field AAG receives an error (AAG_03401) - during set field AAG receives an error (AAG_03404)	502 - bad_gateway - The upstream server response was invalid. (AAG_03401 doesn't comply to that but will be fixed - AR-129818)	Yes, forbidden by JSON schema	AAG_03450 AAG_03451 AAG_03401 AAG_03404
2.2	**In VISS** Upstream server rejects update value **In AAGW** (more specific) - After transformation the value is not accepted by the required service interface it is mapped to (AAG_03452)	502 - bad_gateway - The upstream server response was invalid. Unsure - I would think this is a configuration problem and therefore this error is fine	Yes, forbidden by JSON schema	AAG_03452
2.3	read: in case the retrieved value violates VSS datatype restrictions	502 - bad_gateway - The upstream server response was invalid	yes, but forbidden by JSON schema	AAG_00103,
2.4	subscription: in case the retrieved value violates VSS datatype restrictions	no error, but also no notification		AAG_03383, AAG_03323

Nr	Condition	Error	VISS Transport representation	AR Spec
2.5	Request to optional, unsupported feature e.g. subscribe to VSS branch (only error in certain deployments e.g. AAPI)	501 - not_imlemented - "Update and Subscribe to Branches is not supported"	no, also forbidden by JSON schema	AAG_03322

Errors with no connection to AAPI

- e.g. because they concern filters that are not supported by the AAPI

Nr	Condition	Error	VISS Transport representation	VISS Core Condition(s)
3.1	History filter: if historic data is unavailable	404 - unavailable_data - The requested data was not found Suggestion: make error reason and description more specific for this case	General error is there, something more specific is missing	"if historic data is unavailable."link "A request for historic data will return a Not found error (404)" -> Defines Error code but nothing else

- There are likely more of these situations.