

# Specification

## Make sure you have a clear understanding of the specification

When you are developing an implementation, make sure that you have a clear understanding of the specification of the protocol and the artefact. The specification is meant to be precise enough to ensure that your implementation is interoperable with others or is at least very close to be. In case of questions or unclarities, do not consciously make assumptions but contact the author of the specification. Interoperability is primarily a matter of correct and consensual interpretation of this specification. Assumptions made that are not shared amongst the community of implementers will impede a proper implementation and will require adaptations in a later stage in the software development lifecycle, with a higher cost.

Consider that the intended audience of the specification is wide. Knowing that product managers, end-users, test developers or product developers not always fully share the author's background and expertise, be aware that interpretation mistakes or liberties will abound.

In the special case where two or more specifications refer to each other and compose with each other, the specification writer assumes on some aspects, a "black-box" or perfectly modular composition. This implies that some "corner cases" (error escalation, mismatched features) could not explicitly be documented. This way the details are left to the implementers. The degree of variability introduced by such compositions is then underestimated. As a result, implementers may interpret differently how the composition is supposed to operate. This can be avoided by not making any assumptions in case of doubt, and contact the writer of the specification. E.g.: a Vitalink layer will operate above the eHealth Kmehr artefact specification and Hub webservices protocol specification, these are defined outside of Vitalink.

## Document the used version of the specification

Always choose the latest version of the specification when starting the process of developing a new implementation. Document the version number and make this information accessible for the end-users. Not doing so will cause misunderstandings and interoperability issues over time. Providing the user with clear information about the version and revision numbers of both the implementation and the implemented specification will help to avoid this.

Be aware that it is a very unlikely event that all applications migrate their implementations at the same time to a newer version of the specification.

## Compatibility

When migrating your implementation to a newer version of the specification, make sure that you have identified and implemented mitigation to the non-backward compatible features of the specification. This is needed to be able to correctly consume artefacts produced by [other applications/implementations](#) using the older version(s) of the specification.

## Processing of optional features

The optional character of a specified feature, when concerning an artifact that may be produced and consumed, is a common source of confusion and interoperability failure. An artifact MAY implement a feature. This clearly means that a compliant device producing such an artifact MAY omit this feature. However, any compliant device consuming such an artifact MUST be able to process this feature, should it be present (unless specified otherwise). In this way, useful or important information included in the optional feature, is not omitted by the consuming application.