Versioning Strategy on Profile level

Introduction

Implementation guides are evolving in time. New versions will be published which can contain new business rules and/or validation rules.

We asked advice in the HL7 Belgium FHIR Workgroup Infrastructure & Security.

A proposal document has been started by Bart Decuypere: https://docs.google.com/document/d/1NIaq6NfqrfK46tRTlcGlWEFRrbCzg5uZrs6cgoWZcUs/view

The document recommends strategies based on different dimensions depending on the system and business requirements.

Data storage in Vitalink:

- Vitalink chose to store the FHIR resources as-is with search tables (HAPI) instead of mapping it to an internal datamodel. (Choice was made at the beginning of the project by the architect)

Data integrity in Vitalink:

- We store (medical) data as it comes in. We send the same content back to the client as how it was sent by the creating user (author).
- ? Is the meta property of a resource part of the data integrity?
    - Can the server add the profile and version to the resource before storage?
    - The profile and version is necessary for a receiving client to know which version the resource is in order to parse it correctly.
    - Info of profile/version could be stored outside of the resource as well, but this is not the ideal setup in the Vitalink case and we anyhow need to add this to the resource before it is send to a receiving client. Or will we do transformation ourselves?

Endpoints in Vitalink:

- ? We want to use common endpoints for each FHIR resource type (not more than one for each resource type). This allows clear endpoint definition for integrators. It's more maintainable for Vitalink as the endpoints will remain the same over the whole project duration.

- By defining only one endpoint for each resource type we need know the profile and version of the profile for when a client retrieves the information back later in time (when profiles and versions might have changed).

To summarize the strategy Vitalink:

- Vitalink exposes the StructureDefinitions it supports at that moment on their server (in the supportedProfile part in the CapabilityStatement and by exposing them via the StructureDefinition endpoint).
- For incoming resources
    - with profile and supported version
        - Vitalink will validate agains the rules of that profile and version and store it as-is in the DB
    - with profile and not-supported version
        - Vitalink won't accept it as it's not possible to validate against the rules of the not-supported version of the profile
    - with profile without version
        - Vitalink will validate against the rules of the latest version of the profile (latest as in current version supported by Vitalink, see CapabilityStatement)
        - Vitalink will add the latest version to the meta of the resource so when data is retrieved the client knows which version the resource is compliant to.
    - without profile
        - Vitalink won't accept it. ? Is the profile something we can require? It's not a required field of a FHIR resource type. But as a system we can always add our own rules.
- For outgoing resources
    - The profile and version will always be present as it is checked for incoming messages and added if it's missing.
        - ? What do we do with old versions in the DB? Do we transform it at retrieval time (if we can) or do we send it as-is to the client?

? On the limit of supported versions Vitalink still need to decide on a strategy. Will Vitalink deprecate older versions for reading? Do we know what is expected by clients? ? Assuming Vitalink can't do transformation...

Will Vitalink deprecate older versions for writing? It's best to follow guidelines.

Questions to ask eHealth:

- Will transformation paths be documented by eHealth and is it computable?

Open questions to be answered on standardization matter:

- We can expose StructureDefinitions on the server: but can we say what StructureDefinitions are supported and what not? Or do we only expose the supported StructureDefinitions, like the property in the CapabilityStatement 'supportedProfile'?