Working Document on Revision of IALA G1128

Background:

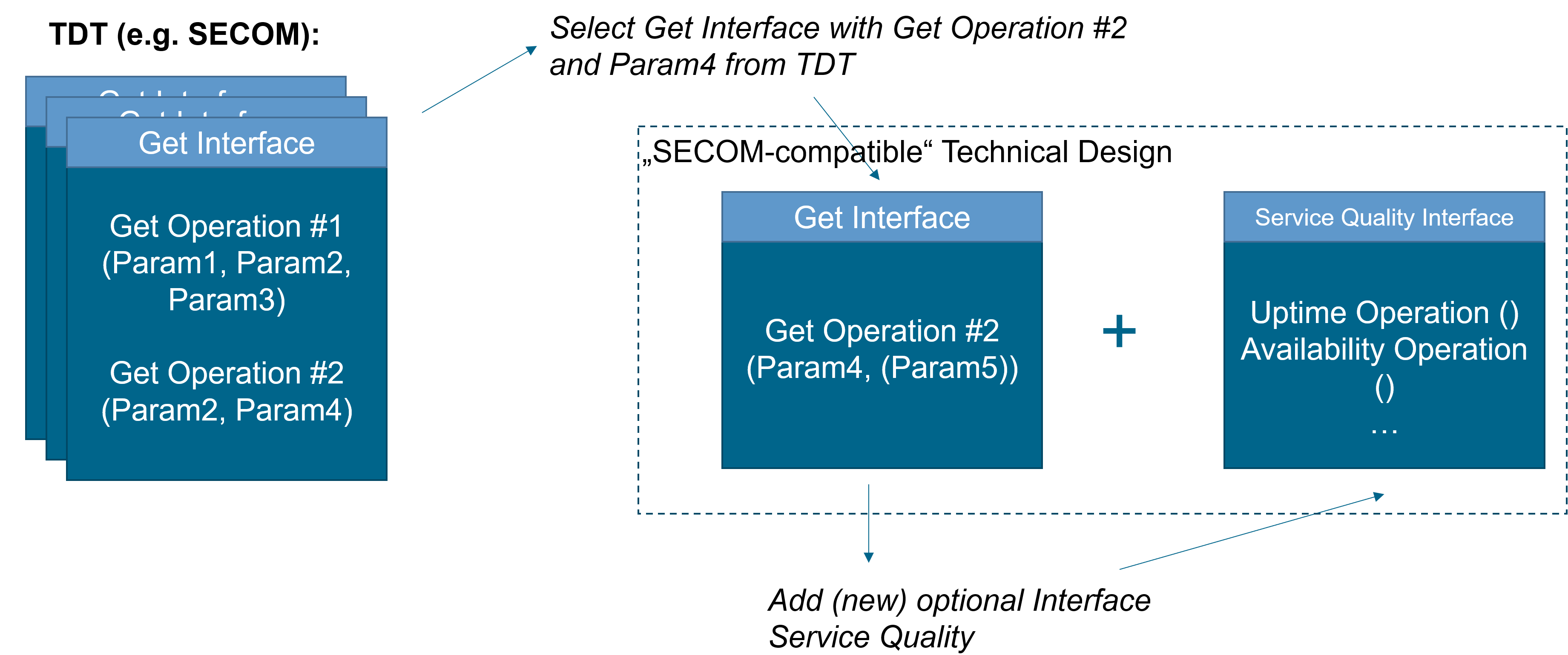
* In the implementation of e-Navigation technical services, it is not uncommon to refer to existing standards for data exchange. The example of a REST-API is adopted and standardised further for the e-Navigation use-case with IEC 63173-2 (SECOM). Another example is using secure file servers (SFTP) to provide S-100 data to mariners.
* On the other hand, a large set of maritime services will be specified in the context of e-Navigation. Developing new descriptions of the same technologies in different technical design specifications is inefficient and can lead to inconsistencies and is a process that is hard to maintain.
* In the worst case, manufacturers of equipment capable of consuming maritime services (e.g., ECDIS manufacturers) would need to implement each technical design of a service individually to support data exchange with that service.
* For this reason, version 2.0 of G1128 introduced the concept of Technical Design Templates.

Definition of Technical Design Template (TDT):

* A Technical Design Template is a collection of interfaces, as specified by the Technical Design data model of G1128.
* The return type of operations in the Interfaces must be abstract types (TDT is payload-data agnostic).
* TDTs can be understood similarly to software libraries: Existing, generic specifications can be “imported” in a technical design and adapted to the existing requirements.
* Utilising TDTs for the development of technical designs has several advantages:
  + Technical Designs can be created in a more efficient and harmonised way, while increasing interoperability.
  + Equipment manufacturers can profit from the harmonisation of technical design specifications with TDTs: Systems might be able to support multiple services without implementing their specific technical design specifications. If a technical design is composed of a TDT and a standardised data model (e.g., S-100 based) and the system implements these to modules, it can support the service.
  + On the other hand, a system could discover the complete set of services it is compatible with on a technology basis. For example, users could search for all services that utilise secure file servers to exchange S-124 data.
* It is recommended to create TDTs for technologies already specified in technical standards or for technologies expected to be utilised in multiple services. For other services, it is still possible to develop technical designs without reference to a TDT.
* The TDT is a versioned document.

Relation to the Technical Design

* The TDT may be utilised for the definition of Technical Design. However, it is not required to do so.
* The relation between TDT and Technical Design is m:n. This means that a technical design may refer to multiple TDTs and that a TDT can be used in various technical designs.
* The TD “imports” a sub-set of interfaces from the TDT.
  + Interfaces may be modified in such a way that only a subset of operations remains in the TD. At least one operation from the TDT must remain in the imported interface.
  + Operations may be modified in such a way that only a subset of parameters remains in the TD.
  + Imported Interfaces may be complemented with additional interfaces. The same is possible for operations inside imported interfaces.
  + Operations and parameters must only be modified inside an interface/operation. It is not possible to combine an operation or parameter with an operation or parameter from a different interface/operation.
  + The TDT can define mandatory interfaces, operations or parameters that must be included in the TD. (Example from Mikael: The same construction are used also in IHO Product Specifications, where certain attributes is optional, but if they are provided they can be either optional or mandatory to process. Let's say the schedule in a route plan; It is optional but if provided, the implementation is expected to have the capability to process the schedule. For a service interface, there can be an Upload interface where you send a Data product, but there is a parameter "Acknowledgement requested". This is optional but if provided, the service must be able to fulfil this request, It is mandatory top process when given.)
  + The selection of interfaces, operations and parameters could be defined with tables inside the technical design. Another possibility is to build services in such a way that the availability of interfaces, operations and parameters can be defined at runtime (details need to be discussed).



* While the Technical Design description document can reference the TDT, the XML file must explicitly define the interfaces without references. The TDT may provide a template that can be used to construct these XML-Files.

Service Discovery via TDT:

* The TD should include a reference to its utilised TDTs.

Further Notes from the discussion:

* Definition of **payload-data format**: The payload-data format refers to a well-defined (or standardized) format, such as the S-100 product family.
* The use of template designs is applicable for several types of S-products, reduces the different types of implementations required (especially important for ship systems) and it quickens the development by re-using existing design template for new services.
* Use of Design Templates reduces the need to inform service consumers that there is now a new design to implement. New services on same template will “pop-up” when searching for services “I am interoperable with”.
* The Design Templates is not mandatory to use. A Service Technical Design may become a future Design Template.
* Design Template is intended to be related to only one technology stack. E.g., Design Template SECOM REST Design is separate from Design Template MMS Design (assuming here that also MMS is a documented design pattern).
* The Technical Design Template is payload-data independent.
* The Service Instance refers to Service Technical Design that refers to zero or many Design Template(s)
* Service Instances are searchable via Design Template (Use-case in a Service Registry)
* Service Instances are searchable via Service Specification (Use-case in a Service Registry)
* The Service Technical Design refers to Service Specification
* The Service Technical Design can include 1 or many payload-data formats. E.g., S-421 and S-124. It is also possible to specify technical designs without a payload-data format, e.g., if a service just acknowledges a user-defined input.
* The scope of Service Specification is complete or part of Use Case and involves both producer and consumer in the interaction patterns and service interfaces. This may mean that a single Service Specification may require an orchestration of several service interfaces. Must then all these interfaces be defined in same Technical Design, or can several Technical Designs fulfil one Service Specification? – It depends on the service architecture: If a clear server-client pattern is applied with the client registering at the server, a single service specification can fulfill the requirements. If the coupling between service client and service provider is not tight (e.g., when other service instances can talk to the same interface on a ship), different service specifications may be utilized for the different entities.

**Open Question**: If a service requires a producer and consumer both to expose interfaces, how is it possible to search for a service endpoint in a service registry based on the service instance description, while differentiating between consumer and producer of the same service design.

Example Use-Case for G1128:

The service specification define input in one payload format and output in another payload format.

E.g. Ship shares its route plan to services and gets weather on route in response.

E.g. Ship shares its route plan to services and gets navigational warnings on route in response.

E.g. Ship shares its route plan to services and gets environmental areas on route in response.

E.g. Ship shares its route plan to services and gets ice on route in response.

The ships has implemented S-421 Route Plan generation in ECDIS and can share this to “S-421 services”. The ship adds parsing and rendering of S-413 and implements service call for S-413.

How many Service Designs are involved? – This depends on the specific services: Both one and multiple services are possible. This is not restricted by G1128.