

## **S-100 – Part XX**

### **Session Oriented Services**



**Contents**

11-1	Scope .....	3
11-2	References .....	3
11-3	General concept of a service .....	3
11-3.1	Session and service components.....	3
11-3.1.1	Model of interactions within a session oriented service.....	3
11-3.1.2	Communication stack. ....	5
11-3.2	Data component.....	5

## 11-1 Scope

This part describes the service components and processes needed to specify an exchange of information of a continuous nature. This is also known as “streaming data”, wherein the data requires a more dynamic information flow to be available, i.e. beyond that found with the exchange of static datasets.

## 11-2 References

- ISO 7498-1:1994. Open Systems Interconnection (OSI) Reference Model.
- IALA Recommendation V-145 “On the Inter-VTS Exchange Format (IVEF) Service” ed. 1 June 2011.
- OGC Sensor Observation Service (<http://www.opengeospatial.org/standards/sos>).

## 11-3 General concept of a service

A service is a collection of functionality and procedures which is provided to its users, under a request/response interaction pattern and/or event driven. The services can be summarized in general as a session oriented or broadcast service. The main distinction between session oriented service and broadcast service is the level of interaction between the service provider and the user of the service. A session oriented service requires interaction between the service provider and the user.

A session oriented service typically contains three components, each handling other types of data:

- Session component: Describing the handling of the session data (service request, service response, login, login response, logout).
- Service component: Describing the information to maintain the service (e.g. keep alive messages, service status).
- Data component: Describing the data itself (e.g. Vessel Traffic Image data (objects))

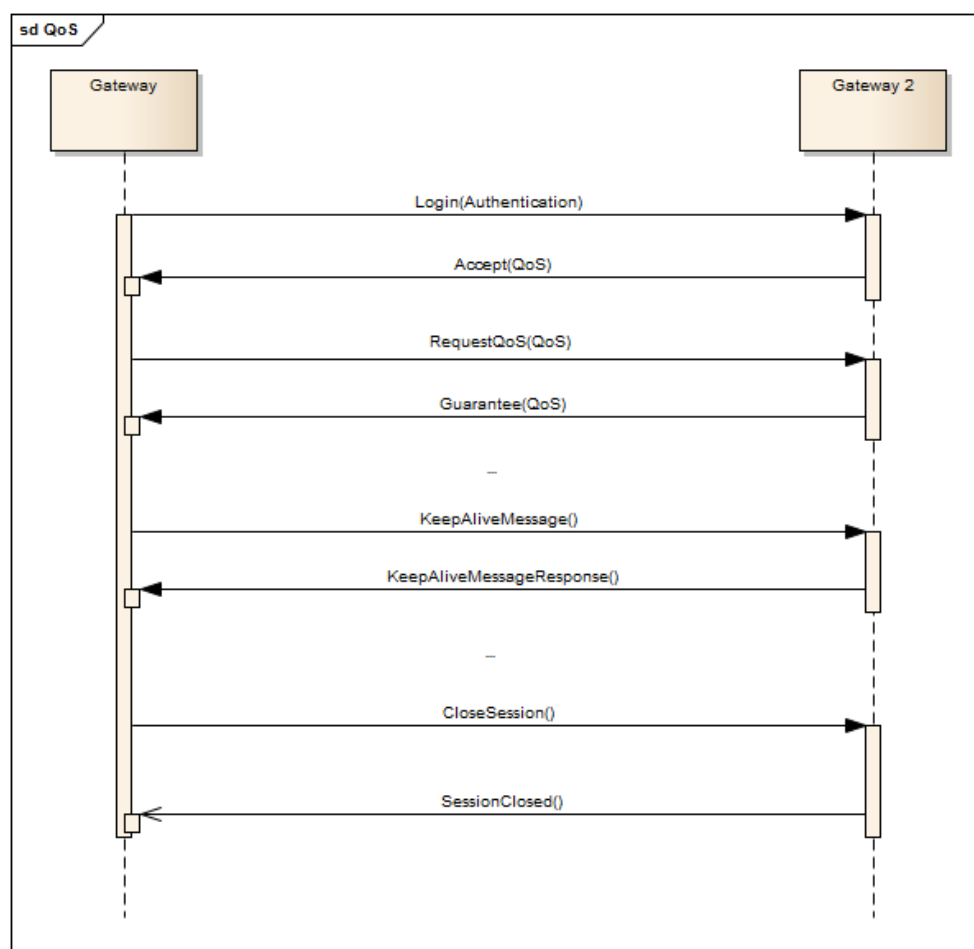
Metadata required for each component should be detailed in the product specification. The metadata should use the concepts and tags as described in S-100 part 4 as much as possible.

### 11-3.1 Session and service components

The session and service components are the “wrappers” around the data components and contains the information to access and maintain the connection with the service.

#### 11-3.1.1 Model of interactions within a session oriented service

In a session oriented service the interfaces are point-to-point connections between Gateways. The Gateways manage the session and exchange information bi-directionally. The service description should contain an interaction model. The interaction model should describe the life span of a session (initiation, maintenance and termination of the session).



*Figure 1 Example of session interaction model*

For each element in the interaction model a detail description must be provided in the product specification of the service. This to ensure that the service interaction is harmonized and reliable. E.g. a description of the protocol used in a service may provide sufficient feedback to ensure full reception of the data, if this is essential for the service.

An example of an interaction model is given in Figure 1.

In the product specification of the service the interactions must be defined. For example the following messages:

- Initiate the session
  - Initiate and confirm Sessions
  - Negotiate the quality of service
  - publish / subscribe information
- Maintenance of Session
  - Keep alive messages
- Termination of the Session
  - Closing Session Request
  - Session Closed Information

Between initiation and termination, both gateways can exchange data messages (synchronous as asynchronous).

### 11-3.1.2 Communication stack.

The means of communication for the use of the service should be defined in a communication stack. Specifying a communication stack will ensure that communication for the service is harmonized and will make implementation easier.

The communication stack should be oriented to ISO-OSI Reference Model and cover for example:

- Session protocols (e.g. WSDL, SOAP, REST, SoS) to define message types
- Encoding and compression (e.g. GML, KML, XML, JSON, ....) to serialize data
- Communication protocol (e.g. HTTP) with encryption (e.g. HTTPS) to define interaction between gateways
- Transportation Layer (e.g. TCP/IP) with encryption (e.g. SSL) to define transportation node between gateways.

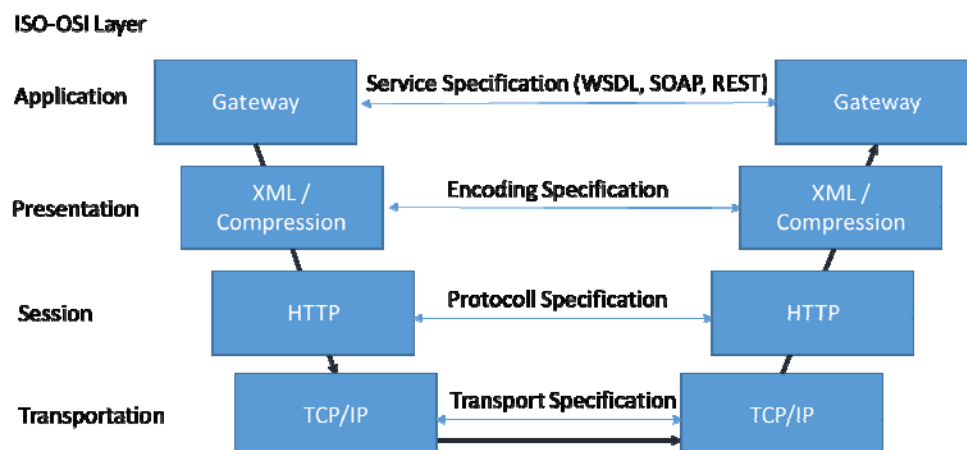


Figure 2 Example of a Communications Stack

### 11-3.2 Data component

The data component describes the data and the structure of the data to be send. The features and attributes are modelled in UML according to S-100 part 3. The features can bundle as dataset and/or messages.