Overview Sensor Web Enablement (SWE)

SWE definition

Set of OGC standards that work together to better:

- discover
- access
- control
- use sensor data.

SWE scope

Let's call all these **Sensor Systems** or *Systems*

- Detectors
- Sensors
- Sensor Networks
- Platforms

SWE motivation

- Systems are disconnected

- Systems are heterogeneous

- Systems produce massive amount of data
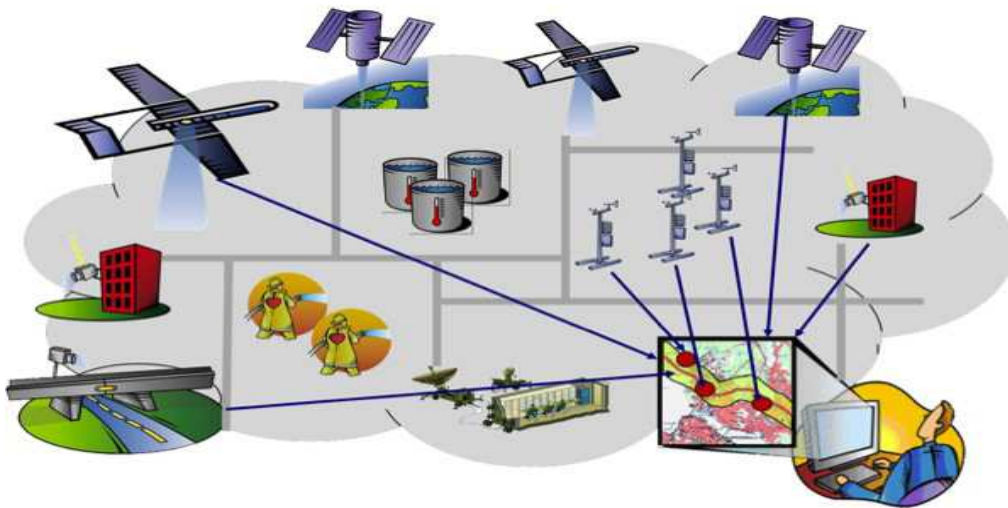
Systems are disconnected

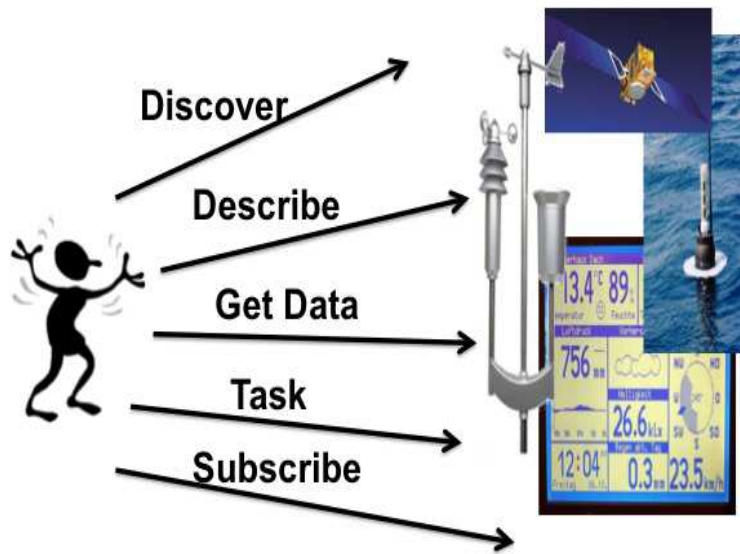# Systems are heterogeneous

Systems produce lots of data

640 terabytes of operational data on just one Atlantic crossing

User connected to sensor systems

SWE requirements

SWE solution

SWE encoding standards

- Observations and Measurements (O&M)
- SensorML
- SWECommon
- PUCK

SWE interface standards

- Sensor Observation Service
- Sensor Planning Service
- PUCK
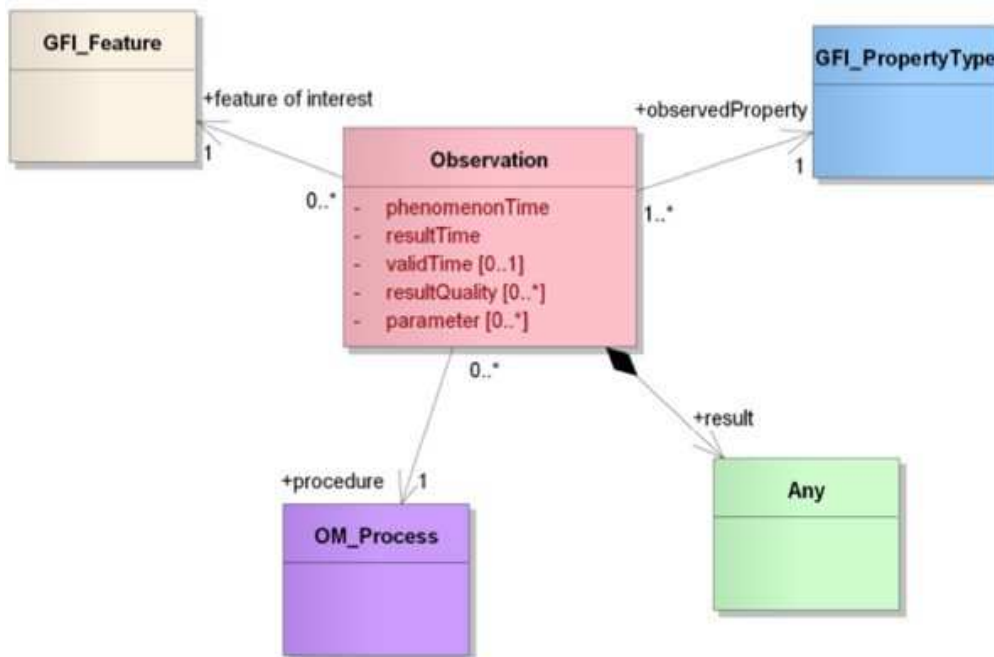
Observations and Measurements (O&M)

This tutorial provides an introduction to the OGC Encoding Standard Observations and Meaurementes

O&M

- An observation is an *event* that
- estimates an *observed property*
- of a *feature of interest*,
- using a *procedure*, and
- generating a *result*

O&M UML

O&M XML Example

Lets walk through an example from Geonovum

O&M XML Header

```xml
<?xml version="1.0" encoding="windows-1250"?>
    <om:Measurement gml:id="obsTest"
        xmlns:om="http://www.opengis.net/om/1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:gml="http://www.opengis.net/gml"
          xsi:schemaLocation="http://www.opengis.net/om/1.0
          ../extensions/observationSpecialization_override.xsd">
```

O&M XML Time

```
<om:samplingTime>
  <gml:TimeInstant>
    <gml:timePosition>2008-10-14T00:09:53+02:00</gml:timePosition>
  </gml:TimeInstant>
</om:samplingTime>
```

O&M XML Procedure

```xml
<om:procedure
    xlink:href=
          "urn:ogc:object:feature:OSIRIS-HWS:
              a1ef1094-c201-4f9f-8f2e-0ff97bf65f03"/>
```

O&M XML Observed Property

```xml
<om:observedProperty
    xlink:href="urn:x-ogc:def:property:OGC::RelativeHumidity"/>
```

O&M XML Feature of Interest

```xml
<om:featureOfInterest>
    <sa:SamplingPoint
  gml:id=
    "urn:ogc:object:feature:OSIRIS-HWS:
        a1ef1094-c201-4f9f-8f2e-0ff97bf65f03"
  xsi:type="ns:SamplingPointType"
        xmlns:ns="http://www.opengis.net/sampling/1.0"
  xmlns:gml="http://www.opengis.net/gml">
  <gml:name>roof of the ifgi</gml:name>
  <sa:sampledFeature
    xlink:href="urn:ogc:object:feature:
        OSIRIS-HWS:hygrometera1ef1094-c201-4f9f-8f2e-0ff97bf65f03"/>
  <sa:position>
    <gml:Point srsName="urn:ogc:def:crs:EPSG:4326">
      <gml:pos>52.07349 9.42125</gml:pos>
    </gml:Point>
  </sa:position>
  </sa:SamplingPoint>
</om:featureOfInterest>
```

O&M XML Observed Property

```xml
<om:result uom="%"
    xlink:href="urn:x-ogc:def:uom:OGC:percent" >41
</om:result>
</om:Measurement>
```

SWECommon

SWECommon provides

- primitive data types (boolean, categories, text, quantities ..)

- aggregate data types (records, arrays, matrices)

- specialized data types (curves, time dependent)

- Structure to encode quality information

- Structure to provide semantic annotations

SWECommon relation with other encoding standards

- In SensorML inputs and outputs are express via SWECommon
- In O&M the result can be expressed with SWECommon

SWECommon XML Example

Namespace declaration:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<swe:DataStream id="EXAMPLE_01"
    xmlns:swe="http://www.opengis.net/swe/2.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xsi:schemaLocation=
        "http://www.opengis.net/swe/2.0 http://schemas.opengis.net/sweCommon/2.0/swe.xsd">
```

SWECommon XML Example

Element Count:

```xml
<swe:elementCount>
    <swe:Count>
        <swe:value>10</swe:value>
    </swe:Count>
</swe:elementCount>
```

SWECommon XML Example

Data Record and definition of first field (Time):

```xml
<swe:elementType name="weather_data">
   <swe:DataRecord>
      <!-- -->
      <swe:field name="time">
         <swe:Time definition=
             "http://www.opengis.net/def/property/OGC/0/SamplingTime">
            <swe:label>Sampling Time</swe:label>
            <swe:uom xlink:href=
                "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"/>
         </swe:Time>
      </swe:field>
```

SWECommon XML Example

Definition of second field (Temperature):

```xml
<swe:field name="temperature">
    <swe:Quantity definition=
          "http://mmisw.org/ont/cf/parameter/air_temperature">
       <swe:label>Air Temperature</swe:label>
       <swe:uom xlink:href="Cel"/>
       <swe:constraint>
          <swe:AllowedValues>
             <swe:value>1</swe:value>
             <swe:value>2</swe:value>
                      <swe:value>3</swe:value>
             <swe:interval>-50 +50</swe:interval>
             <swe:significantFigures>2</swe:significantFigures>
          </swe:AllowedValues>
```

```
        </swe:constraint>
    </swe:Quantity>
</swe:field>
```

SWECommon XML Example

Definition of third field (Pressure):

```xml
<swe:field name="pressure">
    <swe:Quantity definition=
        "http://mmisw.org/ont/cf/parameter/air_pressure">
        <swe:label>Atmospheric Pressure</swe:label>
        <swe:quality>
            <swe:Quantity definition=
                "http://sweet.jpl.nasa.gov/2.0/sciUncertainty.owl#Accuracy">
                <swe:uom code="%"/>
                <swe:value>10</swe:value>
            </swe:Quantity>
        </swe:quality>
        <swe:uom code="mbar"/>
    </swe:Quantity>
</swe:field>
```

SWECommon XML Example

Definition of fourth and fifth fields (Wind Vector):

```xml
<swe:field name="windSpeed">
    <swe:Quantity definition=
            "http://mmisw.org/ont/cf/parameter/wind_speed">
        <swe:uom code="km/h"/>
    </swe:Quantity>
</swe:field>
<!-- -->
<swe:field name="windDirection">
    <swe:Quantity definition=
            "http://mmisw.org/ont/cf/parameter/wind_to_direction">
        <swe:uom code="deg"/>
    </swe:Quantity>
</swe:field>
<!-- -->
        </swe:DataRecord>
    </swe:elementType>
```

SWECommon XML Example

Definition of the encoding and the data:

```xml
<swe:encoding>
    <swe:TextEncoding tokenSeparator=","
            blockSeparator="&#32;"
            decimalSeparator="."/>
</swe:encoding>
<swe:values>
    2009-01-01T10:00:25Z,25.3,1098,5,56
    2009-01-01T10:00:35Z,25.4,1098,15,59
    2009-01-01T10:00:45Z,25.4,1098,12,42
    ...
</swe:values>
</swe:DataStream>
```
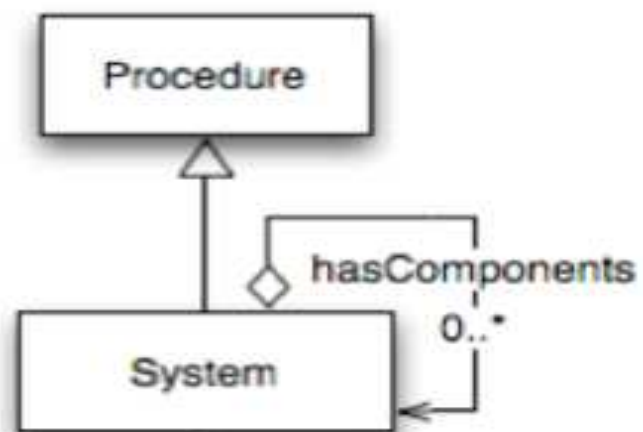
Sensor Model Language (SensorML)
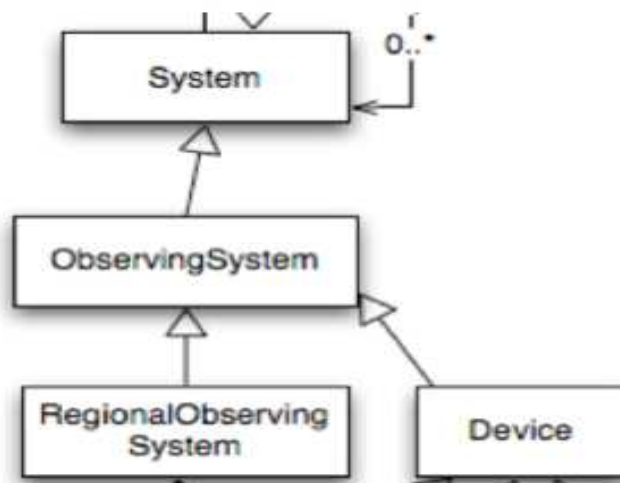
SensorML For Sensor Systems

Model to describe Sensor Systems:

- Detector
- Sensor
- Platform
- Observatories

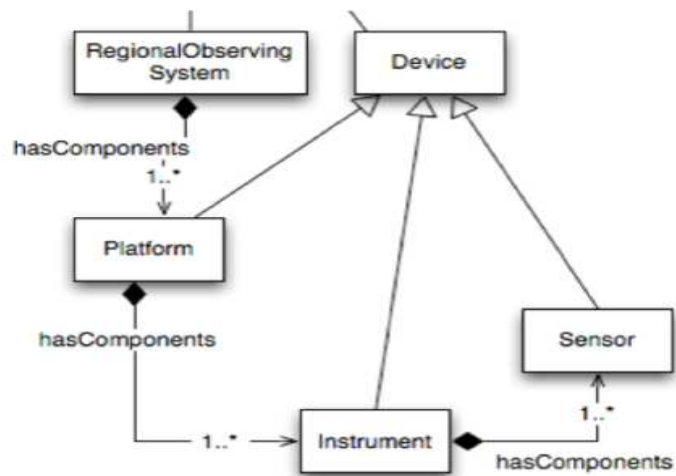# SensorML can define systems

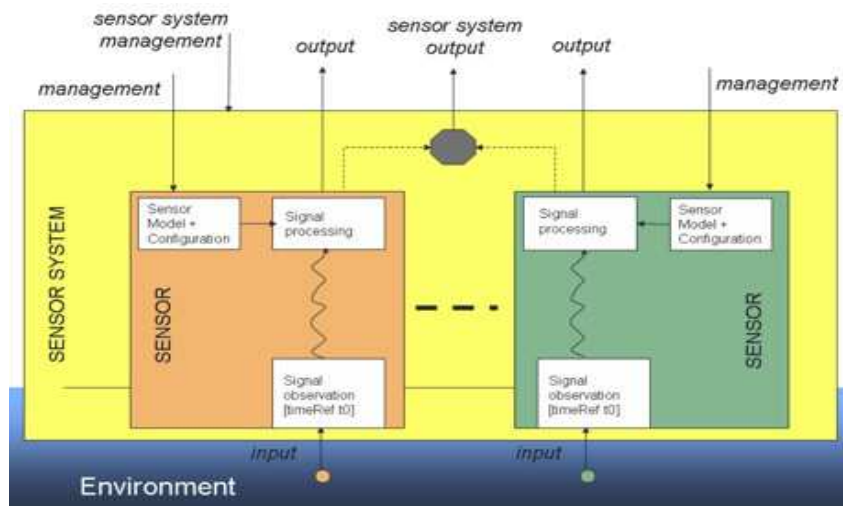# Some types of environmental systems

Instrument composition

SensorML Instrument System
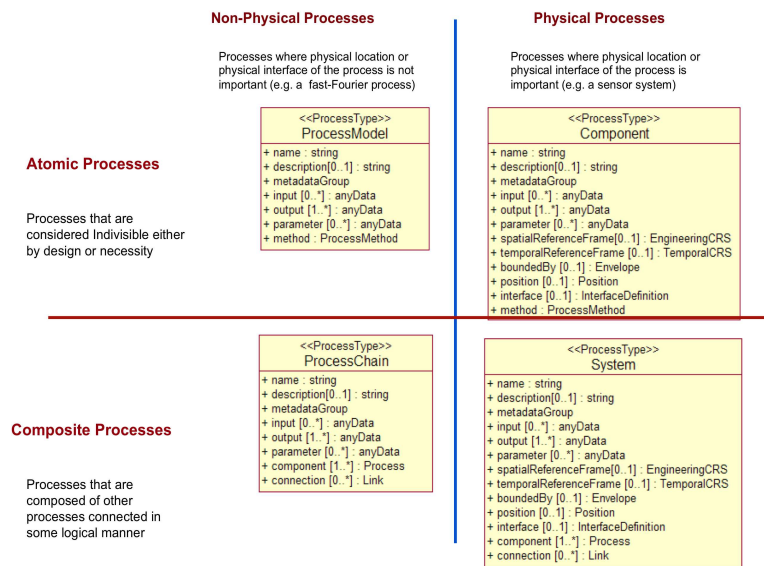

Sensor Web Enablement Architecture, OGC-06-021r4

SensorML can describe a process

Model to describe processes

- inputs
- output
- service chaining

SensorML Processes

Processes where physical location or
physical interface of the process is not
important (e.g. a fast-Fourier process)

Processes where physical location or
physical interface of the process is
important (e.g. a sensor system)

Atomic Processes

Processes that are
considered Indivisible either
by design or necessity

<<ProcessType>>
ProcessModel
+ name : string
+ description[0..1] : string
+ metadataGroup
+ input [0..*] : anyData
+ output [1..*] : anyData
+ parameter [0..*] : anyData
+ method : ProcessMethod

<<ProcessType>>
Component
+ name : string
+ description[0..1] : string
+ metadataGroup
+ input [0..*] : anyData
+ output [1..*] : anyData
+ parameter [0..*] : anyData
+ spatialReferenceFrame[0..1] : EngineeringCRS
+ temporalReferenceFrame [0..1] : TemporalCRS
+ boundedBy [0..1] : Envelope
+ position [0..1] : Position
+ interface [0..1] : InterfaceDefinition
+ method : ProcessMethod

Composite Processes

Processes that are
composed of other
processes connected in
some logical manner

<<ProcessType>>
ProcessChain
+ name : string
+ description[0..1] : string
+ metadataGroup
+ input [0..*] : anyData
+ output [1..*] : anyData
+ parameter [0..*] : anyData
+ component [1..*] : Process
+ connection [0..*] : Link

<<ProcessType>>
System
+ name : string
+ description[0..1] : string
+ metadataGroup
+ input [0..*] : anyData
+ output [1..*] : anyData
+ parameter [0..*] : anyData
+ spatialReferenceFrame[0..1] : EngineeringCRS
+ temporalReferenceFrame [0..1] : TemporalCRS
+ boundedBy [0..1] : Envelope
+ position [0..1] : Position
+ interface [0..1] : InterfaceDefinition
+ component [1..*] : Process
+ connection [0..*] : Link

SensorML XML Example

Lets walk through a SensorML example from Geonovum

SensorML XML Keywords

```xml
<keywords>
  <KeywordList>
    <keyword>weather station</keyword>
    <keyword>precipitation</keyword>
    <keyword>wind speed</keyword>
    <keyword>temperature</keyword>
  </KeywordList>
</keywords>
```

SensorML XML Identification

```xml
<identification>
  <IdentifierList>
    <identifier name="uniqueID">
      <Term definition="urn:ogc:def:identifier:OGC:uniqueID">
        <value>urn:ogc:object:feature:Sensor:IFGI:weatherStation123</value>
      </Term>
    </identifier>
    <identifier name="longName">
      <Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
        <value>OSIRIS weather station 123 on top of the IfGI building</value>
      </Term>
    </identifier>
    <identifier name="shortName">
      <Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
        <value>OSIRIS Weather Station 123</value>
      </Term>
    </identifier>
  </IdentifierList>
</identification>
```

SensorML XML Classification

```xml
<classification>
  <ClassifierList>
    <classifier name="intendedApplication">
      <Term definition="urn:ogc:def:classifier:OGC:1.0:application">
        <value>weather</value>
      </Term>
    </classifier>
  </ClassifierList>
</classification>
```

SensorML XML Valid Time

```xml
<validTime>
  <gml:TimePeriod>
    <gml:beginPosition>2009-01-15</gml:beginPosition>
    <gml:endPosition>2009-01-20</gml:endPosition>
  </gml:TimePeriod>
</validTime>
```

SensorML XML Capabilities

```xml
<capabilities>
  <swe:DataRecord definition="urn:ogc:def:property:capabilities">
    <swe:field name="status">
      <swe:Text definition="urn:ogc:def:property:OGC:1.0:status">
        <gml:description>The operating status of the system.</gml:description>
        <!-- station is active -->
        <swe:value>active</swe:value>
      </swe:Text>
    </swe:field>
    <!-- Area that is observed by the station. In this case is insitu
     It matches the location of the station.  -->
    <swe:field name="observedBBOX">
      <swe:Envelope definition="urn:ogc:def:property:OGC:1.0:observedBBOX">
        <swe:lowerCorner>
```

```xml
          <swe:Vector>
            <swe:coordinate name="easting">
              <swe:Quantity axisID="x">
```

```xml
          <swe:uom code="m"/>
          <swe:value>2592308.332</swe:value>
        </swe:Quantity>
      </swe:coordinate>
      <swe:coordinate name="northing">
        <swe:Quantity axisID="y">
          <swe:uom code="m"/>
          <swe:value>5659592.542</swe:value>
        </swe:Quantity>
      </swe:coordinate>
    </swe:Vector>
  </swe:lowerCorner>
  <swe:upperCorner>
    <swe:Vector>
      <swe:coordinate name="easting">
        <swe:Quantity axisID="x">
          <swe:uom code="m"/>
```

```xml
            <swe:value>2592308.332</swe:value>
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="northing">
          <swe:Quantity axisID="y">
            <swe:uom code="m"/>
            <swe:value>5659592.542</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:upperCorner>
   </swe:Envelope>
  </swe:field>
 </swe:DataRecord>
</capabilities>
```

SensorML XML Contact

```xml
<contact>
  <ResponsibleParty gml:id="WWU_IfGI_weather_station_contact">
    <organizationName>Westfälische Wilhelms-Universität Münster - Sensor Web and Simulation Lab</organizationName>
    <contactInfo>
      <address>
        <electronicMailAddress>swsl-ifgi@listserv.uni-muenster.de</electronicMailAddress>
      </address>
    </contactInfo>
  </ResponsibleParty>
</contact>
```

SensorML Position

```xml
<position name="systemPosition">
  <swe:Position referenceFrame="urn:ogc:def:crs:EPSG:6.14:31466">
    <swe:location>
      <swe:Vector gml:id="SYSTEM_LOCATION">
        <swe:coordinate name="easting">
          <swe:Quantity axisID="x">
            <swe:uom code="m"/>
            <swe:value>2592308.332</swe:value>
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="northing">
          <swe:Quantity axisID="y">
            <swe:uom code="m"/>
            <swe:value>5659592.542</swe:value>
```

```xml
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="altitude">
          <swe:Quantity axisID="z">
            <swe:uom code="m"/>
            <swe:value>297.0</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:location>
  </swe:Position>
</position>
```

SensorML Inputs

```xml
<inputs>
  <InputList>
    <input name="precipitation">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:precipitation"/>
    </input>
    <input name="wind">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:wind"/>
    </input>
    <input name="atmosphericTemperature">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:temperature"/>
    </input>
  </InputList>
</inputs>
```

SensorML Outputs

```xml
<outputs>
  <OutputList>
    <output name="precipitation">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:precipitation">
        <swe:uom code="mm"/>
      </swe:Quantity>
    </output>
    <output name="windDirection">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:windDirection">
        <swe:uom code="deg"/>
      </swe:Quantity>
    </output>
    <output name="windSpeed">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:windSpeed">
```

```xml
        <swe:uom code="m/s"/>
      </swe:Quantity>
    </output>
```

```xml
    <output name="temperature">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:temperature">
        <swe:uom code="Cel"/>
      </swe:Quantity>
    </output>
  </OutputList>
</outputs>
```

SensorML Components

```
<components>
  <ComponentList>
    <component name="rainGauge" xlink:href="http://mySensorMLregistry.com?object=98765"/>
    <component name="anemoneter" xlink:href="http://mySensorMLregistry.com?object=33333"/>
    <component name="thermometer">
      <Component>
        ...
      </Component>
    </component>
  </ComponentList>
</components>
```

SensorML Component

```xml
<!-- similar to System, Contact and Position Information are inherited -->
<Component>
  <keywords>
    ...
  </keywords>
  <identification>
    <IdentifierList>
      <identifier name="uniqueID">
        <Term definition="urn:ogc:def:identifier:OGC:uniqueID">
          <value>urn:ogc:object:feature:Sensor:IFGI:thermometer123</value>
        </Term>
      </identifier>
      <identifier name="longName">
        <Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
```

```xml
          <value>OSIRIS Thermometer at weather station 123</value>
        </Term>
      </identifier>
      <identifier name="shortName">
        <Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
```

```xml
            <value>OSIRIS Thermometer 123</value>
          </Term>
        </identifier>
      </IdentifierList>
    </identification>
    <classification>
      <ClassifierList>
        <classifier name="sensorType">
          <Term definition="urn:ogc:def:classifier:OGC:1.0:sensorType">
            <value>thermometer</value>
          </Term>
        </classifier>
      </ClassifierList>
    </classification>
    <capabilities>
      <swe:DataRecord definition="urn:ogc:def:property:capabilities">
        <swe:field name="status">
```

```xml
          <swe:Text definition="urn:ogc:def:property:OGC:1.0:status">
            <gml:description>The operating status of the system.</gml:description>
            <swe:value>active</swe:value>
          </swe:Text>
        </swe:field>
```

```xml
        </swe:DataRecord>
      </capabilities>
      <inputs>
        <InputList>
          <input name="atmosphericTemperature">
            <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:temperature"/>
          </input>
        </InputList>
      </inputs>
      <outputs>
        <OutputList>
          <output name="temperature">
            <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:temperature">
              <gml:groupName codeSpace="ObservationOffering"> Weather </gml:groupName>
              <swe:uom code="Cel"/>
            </swe:Quantity>
          </output>
        </OutputList>
      </outputs>
    </Component>
```
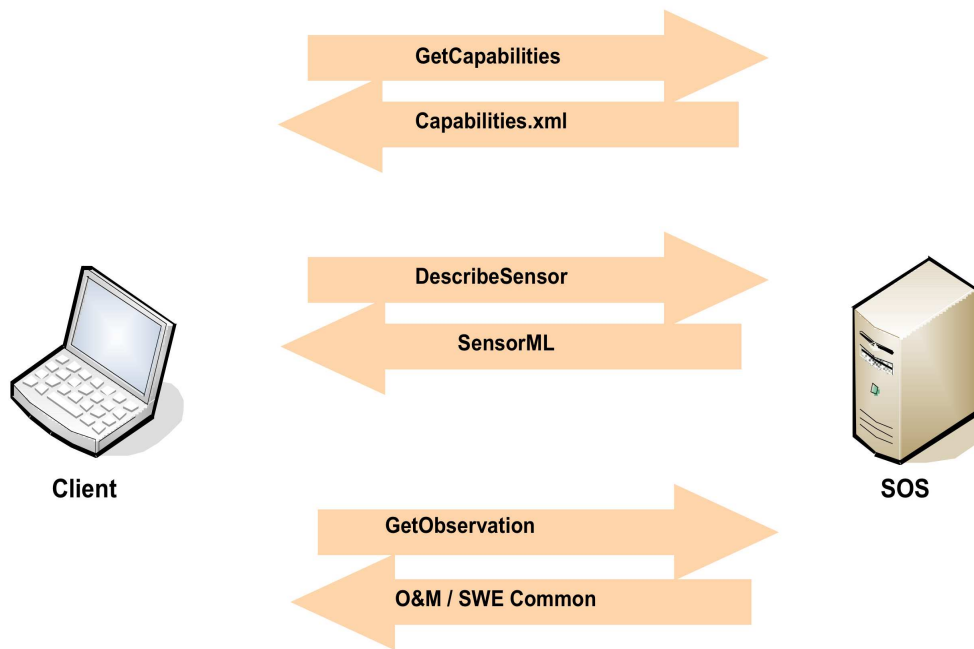
SWE Services

SWE Services

- **Sensor Observation Service** – Access observations and descriptions for sensor systems

- **Sensor Alert Service** – Subscribe to alerts based upon sensor observations

- **Sensor Planning Service** – Request collection feasibility and task sensor system for desired observations

- **Web Notification Service** – Manage message dialogue between client services for long duration (asynchronous) processes

- **Sensor Registries** – Discover sensors and sensor observations

SWE Services
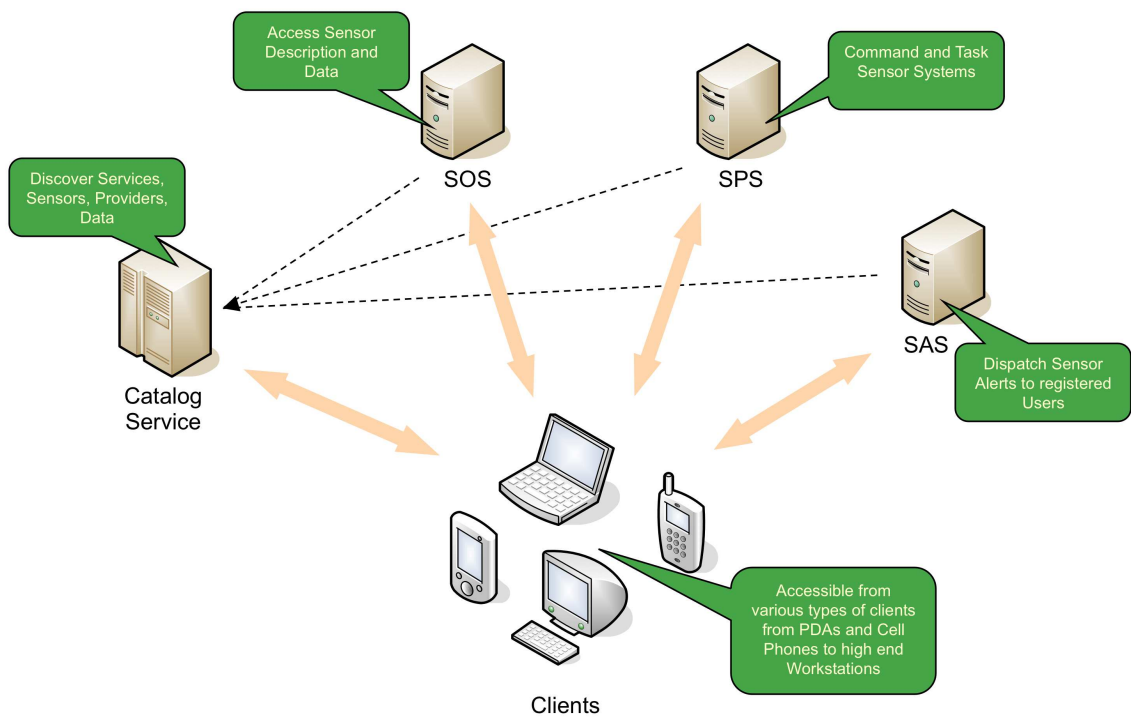
SOS Operations

SOS GetCapabilities

Request example:

```
http://sensorweb.demo.52north.org/52nSOSv3.2.1/sos?
request=GetCapabilities&
version=1.0.0&
service=SOS
```

[Link](#)

SOS DescribeSensor

Request example:

```
http://sensorweb.demo.52north.org/52nSOSv3.2.1/sos?
request=DescribeSensor&
version=1.0.0&
service=SOS&
procedure=urn:ogc:object:feature:Sensor:IFGI:ifgi-sensor-1&
outputFormat=text/xml;subtype="sensorML/1.0.1
```

SOS GetObservation

Request example:

```
http://sensorweb.demo.52north.org/52nSOSv3.2.1/sos?
request=GetObservation&
version=1.0.0&
service=SOS&
offering=GAUGE_HEIGHT&
procedure=urn:ogc:object:feature:Sensor:IFGI:ifgi-sensor-1&
observedProperty=urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel&
resPonSeFormat=text/xml;subtype="om/1.0.0
```
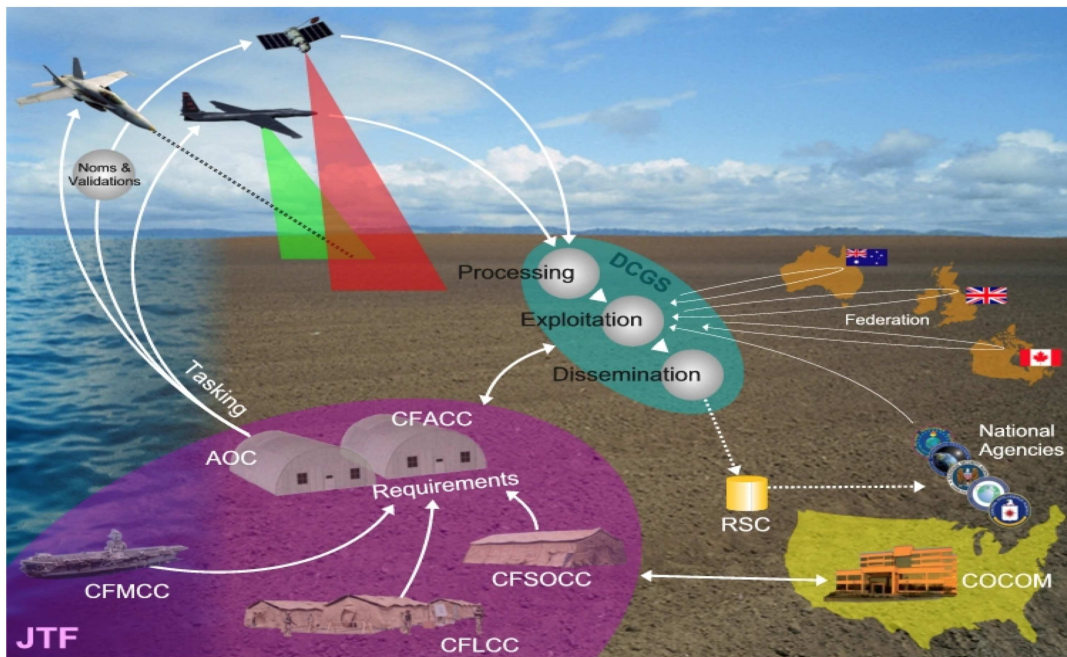
SWE By National Data Buoy Center

SWE and Geo-processing Workflow
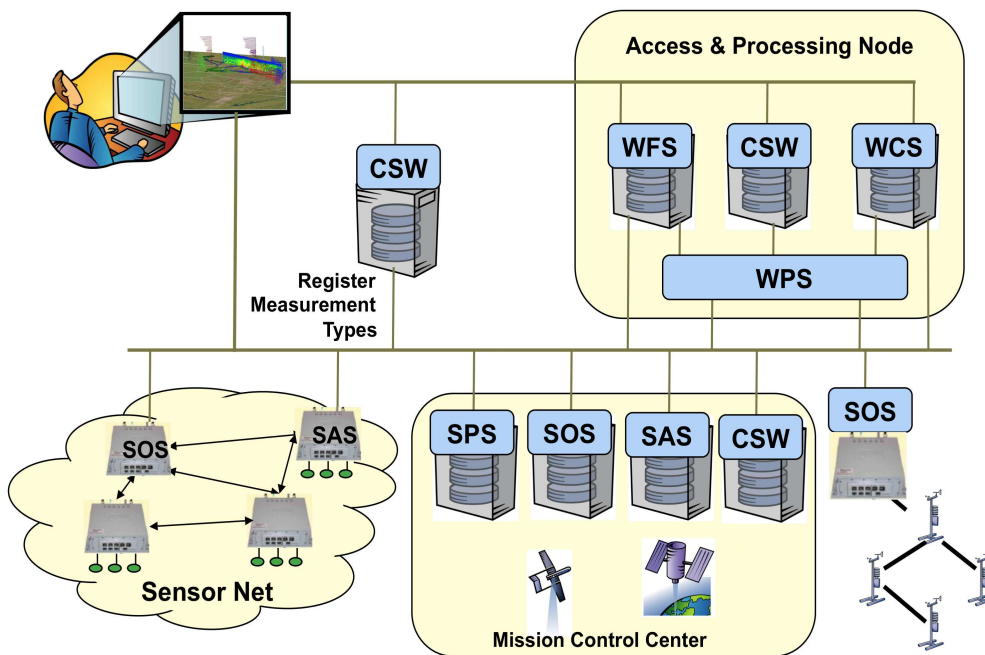
SWE in Defense

SWE in Air Quality