The Annual Conference on Management and Information Technology (ACMIT) 2014

**WS-Security on PRODML**

*Bobby Suryajaya*  
*Master of Information Technology, Swiss German University, Jakarta, Indonesia*  
*Bobby.Suryajaya@gmail.com*

**Abstract**

SKK Migas plans to apply end-to-end security based on Web Services Security (WS-Security) for Sistem Operasi Terpadu (SOT). However, there are no prototype or simulation results that can support the plan that has already been communicated to many parties. This paper proposes an experiment that performs PRODML data transfer using WS-Security by altering the WSDL to include encryption and digital signature. The experiment utilizes SoapUI, and successfully loaded altered PRODML WSDL that had been altered with WSP-Policy based on X.509 to transfer a SOAP message.

*Keywords: SKK Migas, SOT, Web Services, WS-Security, WSDL, WSP-Policy, X.509, SOAP message*

### 1. Background

XML schemas convey the data syntax and semantics for various application domains, such as Oil and Gas production status reports. However, these schemas seldom address security issues, which can lead to a worst-case scenario of systems and protocols with no security at all [5].

SKK Migas plans to apply end-to-end security based on Web Services Security (WS-Security) for Sistem Operasi Terpadu (SOT). Figure 1 describes how WS-Security will be implemented, which employs encryption and signing based on X.509 or Username Token.

![Figure 1. WS-Security Implementation Plan [8]](image-url)
2. Problem Statement

SKK Migas intention’s to apply end-to-end security to existing SKK Migas SOT infrastructure has been communicated to many parties, including vendors. Ideally, prior to communicating to other parties, SKK Migas should have already had a plan on how to implement the idea. Furthermore a prototype or simulation must be ready to support their intention.

The fact is SKK Migas does not have any infrastructure or environment to simulate the idea; hence, there are no prototype or simulation results that can support the plan. But there should be a way to prove whether the architecture can be implemented correctly.

This research proposes an experiment that can simulate existing SKK Migas SOT architecture to utilize WS-Security on SOT data exchange standard, PRODML. This research will perform modification to initial WSDL that was created to transfer data in PRODML format, to match WS-Security requirements, and test the modified WSDL by loading it with a web service simulation tool called SoapUI.

3. Production Markup Language (PRODML)

PRODML is an industry initiative for exchanging production-oriented data in upstream Oil & Gas, including standard data structure in XML format through web services [2].

Few major energy companies initiated PRODML in year 2005 and the initiatives were facilitated and maintained by Energistics. Development of PRODML follows successful WITSML adoption to the Oil & Gas industry as standard for drilling and completion of information architecture. International and national companies have implemented this technology to bring benefits and to add value to business process [1].

According to Weltevrede research, PRODML is a tool that enables optimization and reporting architectures and data management processes to adapt to changes in production environment faster with less effort and fewer errors, which can be used in implementing robust, trustworthy optimization and automation processes [10].

PRODML is adopted by SKK Migas as SOT first phase standard data exchange for Production and Lifting Report from the PSC Contractors as the Operator of the Upstream Oil & Gas assets.

There are four PRODML data-schema required by SKK Migas within SOT implementation for production data. They are: Product Flow Model (PFM), Product Volume Report (PV), Production Operation Report (PO), and Well Test Report (WT). Each data-schema consists of many objects and each object can be generated into a single XML file.

This research uses PRODML PFM to generate a SOAP message during the experiment.

4. WSDL and WS-Security

WSDL or Web Services Description Language is a web services interface description written in machine programmable format. WSDL defines parameters or data structures to serve the data [9].

WS-Security is a flexible and feature-rich extension to SOAP to apply security to web services. WSS specification leveraged SOAP foundation layer such as SOAP, WSDL, XML encryption, XML signature, and SSL/TLS. In this research, Encryption & Signing to PRODML WSDL using X.509 will be employed during data transfer [8].
5. X.509 Profile

X.509 is an ITU-T standard for Public Key Infrastructure (PKI). It can be used by WS-Security as a security token to ensure confidentiality and integrity of a message. X.509 specifies standard formats for public key certificates, certificate revocation list, attribute certificates, and a certification path validation algorithm [3].

Figure 2 describes the scenario on how X.509 will be implemented to provide confidentiality and integrity to data transfer through modification of PRODML WSDL. It is assumed that Certificate Authority is available to hold client and server public certificates.

6. The Testing Tool – SoapUI

The testing here means the process of executing SOAP web services that use XML language for definition of message architecture and message format, which will be in the form of WSDL. There are many tools available in the market. The core functions are the same but they are different in functionality, features, usability, and interoperability. There are several open-source testing tools available in the market, including Jmeter, Storm and SoapUI. This research will utilize, SoapUI.

SoapUI is a free and open source cross-platform Functional Testing solution. It has an easy-to-use graphical interface and enterprise-class features which allows SoapUI users to easily and rapidly create and execute automated functional, regression, compliance, and load tests. In a single test environment, SoapUI provides complete test coverage and supports all the standard protocols and technologies [7].

Beside free and cross-platform, consideration to choose SoapUI is listed as follows:

- SoapUI has complete and automated testing solution.
- SoapUI has easy-to-use graphical interface to work with SOAP-based web services.
- SoapUI has Mock-Services feature that gives unique ability to mimic Web services and create/run Functional and Load Tests against them.
- SoapUI has powerful and flexible reporting tools: Printable, Data Export, and HTML Reports.

This research utilizes SoapUI Pro Version 4.5.2 for MacOSX Version 10.8.4 with 15-days Trial License.

Figure 2. X.509 Implementation Scenario according to WSO2 ESB
7. Research Experiment

A. PRODML WSDL

Basically, normal PRODML WSDL (detail of PRODML WSDL can be found on Appendix) has following structure as defined by Energistics:

1. WSDL definitions
2. WSDL types (including XSD schema)
3. WSDL message
4. WSDL portType
5. WSDL binding
6. WSDL service

In normal deployment, this WSDL can be published directly to web service registry. However, related to SKK Migas SOT implementation, this WSDL must be binding with WS-Security as defined in the implementation requirement document. Therefore, it must be modified to incorporate the security mechanism.

For WS-Security to work there will be another party involved in the transaction, which are key store or certificate publisher and the key itself. The functionality of certificate publisher is to provide certificates that contain key to encrypt / decrypt soap messages, and to perform digital signing of messages and verification of its digital signature.

Implementation of WS-Security will append following structure to the original PRODML WSDL:

1. WSP Policy
2. WSP PolicyReference.

B. WSDL Load Test on SoapUI

Following is one test that was performed on PRODML WSDL with an operation called GetCapabilities.

GetCapabilities Request
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <gen:GetCapabilities/>
   </soapenv:Body>
</soapenv:Envelope>

GetCapabilities Response
<soapenv:Envelope
xmlns:gen="http://www.prodml.org/api/210/genericDataAccess"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <GenericDataAccessCapabilities
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:gml="http://www.opengis.net/gml/3.2"
         xmlns:xlink="http://www.w3.org/1999/xlink">
         <SupportedDataObjects
            xmlns="http://www.prodml.org/api/210/genericDataAccess">
            <SupportedDataObject>
<Name>productFlowModels</Name>
</SupportedOperation>
</SupportedDataObject>

<Namename>productVolumes</Name>
</SupportedOperation>
</SupportedDataObject>

<Namename>wellTests</Name>
</SupportedOperation>
</SupportedDataObject>

<Namename>productionOperations</Name>
</SupportedOperation>
</SupportedDataObject>

</SupportedDataObjects>
</Properties>
<Properties
xmlns="http://www.prodm1.org/api/210/genericDataAccess">
  <NameValuePair>
    <Name>Name</Name>
    <Value>Server #1</Value>
  </NameValuePair>
  <NameValuePair>
    <Name>Version</Name>
    <Value>1.0.0.1</Value>
  </NameValuePair>
  <NameValuePair>
    <Name>Vendor</Name>
    <Value>BIA Energi</Value>
  </NameValuePair>
</Properties>
</GenericDataAccessCapabilities>
</soapenv:Body>
</soapenv:Envelope>

C. WSP Policy

<wsp:Policy xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:AsymmetricBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
      <wsp:Policy>
        <sp:InitiatorToken>
          <wsp:Policy>
            <sp:X509Token
            sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/In-
cludeToken/AlwaysToRecipient">
              <wsp:Policy>
                <sp:RequireThumbprintReference />
                <sp:WssX509V3Token10 />
              </wsp:Policy>
            </sp:X509Token>
          </wsp:Policy>
        </sp:InitiatorToken>
      </wsp:Policy>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
<wsp:Policy>
<sp:X509Token
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/In
cludeToken/Never">
<wsp:Policy>
<sp:RequireThumbprintReference />
<sp:WssX509V3Token10 />
</wsp:Policy>
</sp:X509Token>
</wsp:Policy>
</sp:RecipientToken>
<sp:AlgorithmSuite>
<wsp:Policy>
<sp:Basic256 />
</wsp:Policy>
</sp:AlgorithmSuite>
</sp:Layout>
<wsp:Policy>
<sp:Strict />
</wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp />
<sp:OnlySignEntireHeadersAndBody />
</wsp:Policy>
</sp:AsymmetricBinding>
<sp:AlgorithmSuite>
<sp:Policy>
<sp:Basic256 />
</sp:Policy>
</sp:AlgorithmSuite>
</wsp:Policy>
</sp:AlgorithmSuite>
</wsp:Policy>
</sp:Wss11>
</wsp:Policy>
</sp:SignedParts>
<sp:EncryptedParts
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
>
<sp:Body />
</sp:EncryptedParts>
</wsp:ExactlyOne>
</wsp:Policy>

D. WSP PolicyReference

<wSDL:binding name="PROD_GenericDataAccessSoap"
type="tns:PROD_GenericDataAccessSoap"
8. Result

WSP Policy and WSP PolicyReference were appended to original PRODML WSDL. The WSDL script has included X.509 token that referring to xmlsoap.org schema. The result was tested with SoapUI by loading whole XML script.

SoapUI Pro is equipped with XML-check capability, and the result come up with successful Mock Service loading and starting of modified PRODML WSDL. However, SoapUI functionality test cannot be done, since it needs more effort to create a testing script to utilize SoapUI GetCapabilities and PutData operation, and this can be considered as future work improvement to this research.

9. Conclusion

Based on the experiment results, it is concluded that:

- WS-Security as one of web services security mechanism is appropriate to secure PRODML data-exchange. Successful loading of PRODML WSDL that has been added with WSP-Policy for X.509 as well as mock-service initiation verifies the conclusion.
- By adding WS-Policy that uses X.509 Profile to PRODML WSDL, SKK Migas SOA architecture should have encryption and digital signing capability; hence; providing confidentiality, integrity, availability, authentication, and non-repudiation of the whole interoperability using PRODML.
- WS-Security works by performing encryption and digital signing to each SOAP message. This might lead to performance issues when dealing with enterprise interoperability and growth of transactions. Another mechanism called WS-Secure Conversation might be able to help answering this issue later.

10. Recommendation

There are many improvements that can be done to make this research better as listed below:

- If time permits, conduct better research preparation from all aspects that may include better data preparation (PRODML PFM, PV, PO, and WT) with more detailed documentation on the testing scenario.
- Perform manual functionality test on SoapUI, specifically when dealing with WS-Security, since automated testing is not provided, at least to the knowledge of the author. This includes test-script for GetCapabilities and PutData operation.
- Even though SoapUI Pro has mock services to simulate web services, it is better to use a real orchestration tool such as Enterprise Service Bus (ESB) since mock service has limited capability to the knowledge of the author. There are many open sources ESB where one of them that considered adequate for research is WSO2 Enterprise Service Bus. WSO2 supports security mechanism such as WS-Security.
- Expand the research with other security mechanism such as WS-Secure Conversation or other mechanism that together builds web service security management model.
References


11. Appendix – PRODML WSDL

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions

  name="GenericDataAccess"

targetNamespace="http://www.prodml.org/api/210/genericDataAccess"
xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.prodml.org/api/210/genericDataAccess"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsd/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsd/

xmlns:abs="http://www.energistics.org/schemas/abstract">
  <!-- -->
</!-- Energistics License Agreement
```
This file is distributed under the Energistics License Agreement at http://www.energistics.org
Use of this file constitutes agreement with the Energistics License Agreement.
Copyright (c) 2008-2011 Energistics. All rights reserved.
Energistics, WITSML, PRODML and RESQML are trademarks or registered trademarks of Energistics.

<xsd:schema targetNamespace="http://www.prodml.org/api/210/genericDataAccess" elementFormDefault="qualified">
    <xsd:complexType name="NameValuePair">
        <xsd:annotation>
            <xsd:documentation>A generic structure for option values that are not specified in schema.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Name" type="xsd:string" minOccurs="1" minOccurs="1" />
            <xsd:element name="Value" type="xsd:string" minOccurs="1" minOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayOfNameValuePair">
        <xsd:annotation>
            <xsd:documentation>A collection of generic option values.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="NameValuePair" type="tns:NameValuePair" minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="PutDataResultStatus">
        <xsd:annotation>
            <xsd:documentation>An enumeration which specifies the status of a PutData operation with a data-object.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Added" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
<xsd:enumeration value="Updated" />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ArrayOfSupportedDataObject">
  <xsd:annotation>
    <xsd:documentation>A collection of SupportedDataObject objects.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="SupportedDataObject" type="tns:SupportedDataObject" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SupportedDataObject">
  <xsd:annotation>
    <xsd:documentation>Describes a single element or data type supported by a GenericDataAccess server.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string" minOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>A string describing the version of the type. In the case of prodml/wtsml objects, this must exactly match the version attribute found in the top-level plural object.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="namespace" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:simpleType name="SupportedOperation">
  <xsd:annotation>
    <xsd:documentation>Enumeration which indicates the operations supported on a given data type for a GDA server. A server can support PutData, GetData, or both.</xsd:documentation>
  </xsd:annotation>
</xsd:simpleType>
<xsd:annotation>
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="GetData" />
  <xsd:enumeration value="PutData" />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="GenericDataAccessCapabilities">
  <xsd:annotation>
    <xsd:documentation>Describes the capabilities of this server.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="SupportedDataObjects" type="tns:ArrayOfSupportedDataObject" />
    <xsd:element name="Properties" type="tns:ArrayOfNameValuePair" />
  </xsd:sequence>
</xsd:complexType>

<!-- GetCapabilities -->
<xsd:element name="GetCapabilities">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for GetCapabilities</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence />
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCapabilitiesResponse">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for response from GetCapabilities</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="getCapabilitiesResult" type="tns:GenericDataAccessCapabilities" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- PutData -->
<xsd:complexType name="PutDataResult">
  <xsd:annotation>
    <xsd:documentation>Contains information regarding the response to a DataOperation outside of the returned data itself.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Status" type="tns:PutDataResultStatus" minOccurs="1" />
    <xsd:element name="Id" type="xsd:string" minOccurs="1" />
    <xsd:element name="SuppMsg" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<p>PutData</p><xsd:documentation>SOAP Message for PutData</xsd:documentation>
</xsd:complexType>

<p>PutDataResponse</p><xsd:documentation>SOAP Message for response from PutData</xsd:documentation>
</xsd:complexType>

<!-- Faults -->
<xsd:element name="UnsupportedOptionFault">
  <xsd:complexType>
    <xsd:sequence>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="OptionName" type="xsd:string" />

</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="UnsupportedDataObjectFault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="DataObjectName" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="InvalidXPathFault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="XPath" type="xsd:string" />
      <xsd:element name="XPathProcessorError" type="xsd:string" />
      <xsd:element name="Position" type="xsd:integer" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="UnknownUidFault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Uid" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="UnsupportedCriterionFault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="CriterionName" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="MissingParameterFault">
  <xsd:complexType>
  </xsd:complexType>
</xsd:element>
<xsd:sequence>
  <xsd:element name="Parameter" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>

<!-- GetCapabilities -->
<wsdl:message name="GetCapabilitiesSoapIn">
  <wsdl:part name="parameters" element="tns:GetCapabilities" />
</wsdl:message>
<wsdl:message name="GetCapabilitiesSoapOut">
  <wsdl:part name="parameters" element="tns:GetCapabilitiesResponse" />
</wsdl:message>

<!-- PutData -->
<wsdl:message name="PutDataSoapIn">
  <wsdl:part name="parameters" element="tns:PutData" />
</wsdl:message>
<wsdl:message name="PutDataSoapOut">
  <wsdl:part name="parameters" element="tns:PutDataResponse" />
</wsdl:message>

<!-- Faults -->
<wsdl:message name="MissingParameterFaultMessage">
  <wsdl:part name="fault" element="tns:MissingParameterFault" />
</wsdl:message>
<wsdl:message name="UnsupportedOptionFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedOptionFault" />
</wsdl:message>
<wsdl:message name="UnsupportedDataObjectFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedDataObjectFault" />
</wsdl:message>
<wsdl:message name="InvalidXPathFaultMessage">
  <wsdl:part name="fault" element="tns:InvalidXPathFault" />
</wsdl:message>
<wsdl:message name="UnknownUidFaultMessage">
  <wsdl:part name="fault" element="tns:UnknownUidFault" />
</wsdl:message>
<wsdl:message name="UnsupportedCriterionFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedCriterionFault" />
</wsdl:message>
<wsdl:part name="fault" element="tns:UnsupportedCriterionFault" />
</wsdl:message>
<wsdl:portType name="PROD_GenericDataAccessSoap">
<wsdl:operation name="GetCapabilities">
<wsdl:input message="tns:GetCapabilitiesSoapIn" wsaw:Action="http://www.prodml.org/api/210/genericDataAccess/GetCapabilities" />
</wsdl:operation>
<wsdl:operation name="PutData">
<wsdl:input message="tns:PutDataSoapIn" wsaw:Action="http://www.prodml.org/api/210/genericDataAccess/PutData" />
<wsdl:fault name="UnsupportedOptionFault" message="tns:UnsupportedOptionFaultMessage" />
<wsdl:fault name="UnsupportedDataObjectFault" message="tns:UnsupportedDataObjectFaultMessage" />
<wsdl:fault name="UnknownUidFault" message="tns:UnknownUidFaultMessage" />
<wsdl:fault name="MissingParameterFault" message="tns:MissingParameterFaultMessage" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PROD_GenericDataAccessSoap" type="tns:PROD_GenericDataAccessSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
<!-- GetCapabilities Operation -->
<wsdl:operation name="GetCapabilities" soapAction="http://www.prodml.org/api/210/genericDataAccess/GetCapabilities" style="document" />
<wsdl:input>
    <soap:body use="literal" />
</wsdl:input>
<wsdl:output>
    <soap:body use="literal" />
</wsdl:output>
<!-- PutData Operation -->
<wsdl:operation name="PutData" soapAction="http://www.prodml.org/api/210/genericDataAccess/PutData" style="document" />
<wsdl:input>
    <soap:body use="literal" />
</wsdl:input>
12. Appendix – PRODML WSDL with X.509 Token Profile

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  name="GenericDataAccess"
targetNamespace="http://www.prodml.org/api/210/genericDataAccess"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.prodml.org/api/210/genericDataAccess"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:abs="http://www.energistics.org/schemas/abstract">
</wsdl:definitions>
<wsp:Policy wsu:Id="SigEncr"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:AsymmetricBinding
        xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:InitiatorToken>
            <wsp:Policy>
              <sp:X509Token
                sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
                <wsp:Policy>
                  <sp:RequireThumbprintReference />
                </wsp:Policy>
                <sp:WssX509V3Token10 />
              </sp:X509Token>
            </wsp:Policy>
          </sp:InitiatorToken>
          <sp:RecipientToken>
            <wsp:Policy>
              <sp:X509Token
                sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/Never">
                <wsp:Policy>
                  <sp:RequireThumbprintReference />
                </wsp:Policy>
                <sp:WssX509V3Token10 />
              </sp:X509Token>
            </wsp:Policy>
          </sp:RecipientToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic256 />
            </wsp:Policy>
          </sp:AlgorithmSuite>
        </wsp:Policy>
      </sp:AsymmetricBinding>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
An annual Conference on Management and Information Technology 2014
Master of Information Technology Department, Faculty of Engineering and Information Technology, SGU

</wsp:ExactlyOne>
</wsp:Policy>

<!-- Energistics License Agreement
This file is distributed under the Energistics License Agreement at
http://www.energistics.org
Use of this file constitutes agreement with the Energistics License Agreement.
Copyright (c) 2008-2011 Energistics. All rights reserved.
Energistics, WITSML, PRODML and RESQML are trademarks or registered trademarks of Energistics.
-->
<!--
<wsp:ExactlyOne>
</wsp:Policy>

<wsdl:types>
<xsd:schema targetNamespace="http://www.prodml.org/api/210/genericDataAccess"
   elementFormDefault="qualified">
   <xsd:import namespace="http://www.energistics.org/schemas/abstract"
schemaLocation="http://w3.energistics.org/schema/abstract_v1.0/xsd_schemas/sub_abstractSubstitutionGroup.xsd" />

   <xsd:complexType name="NameValuePair">
      <xsd:annotation>
         <xsd:documentation>A generic structure for option values that are not specified in schema.</xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
         <xsd:element name="Name" type="xsd:string" minOccurs="1" />
         <xsd:element name="Value" type="xsd:string" minOccurs="1" />
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="ArrayOfNameValuePair">
      <xsd:annotation>
         <xsd:documentation>A collection of generic option values.</xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
         <xsd:element name="NameValuePair" type="tns:NameValuePair" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="PutDataResultStatus">
      <xsd:annotation>
         <xsd:documentation>An enumeration which specifies the status of a PutData operation with a data-object.</xsd:documentation>
      </xsd:annotation>
   </xsd:simpleType>
</xsd:schema>
</wsdl:types>
<xsd:complexType name="ArrayOfSupportedDataObject">
  <xsd:annotation>
    <xsd:documentation>A collection of SupportedDataObject objects.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="SupportedDataObject" type="tns:SupportedDataObject" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SupportedDataObject">
  <xsd:annotation>
    <xsd:documentation>Describes a single element or data type supported by a GenericDataAccess server.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string" minOccurs="1">
      <xsd:annotation>
        <xsd:documentation>The name of the type. This should be exactly the same as the local name (sans namespace) of the supported element or type.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="SupportedOperation" type="tns:SupportedOperation" minOccurs="0" maxOccurs="2" />
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>A string describing the version of the type. In the case of prodml/witsml objects, this much exactly match the version attribute found in the top-level plural object.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="namespace" type="xsd:string" use="optional" />
</xsd:complexType>

<xsd:simpleType name="SupportedOperation">
</xsd:simpleType>
<xsd:annotation>
  <xsd:documentation>Enumeration which indicates the operations supported on a given data type for a GDA server. A server can support PutData, GetData, or both.</xsd:documentation>
</xsd:annotation>

<xsd:restriction base="xsd:string">
  <xsd:enumeration value="GetData" />
  <xsd:enumeration value="PutData" />
</xsd:restriction>

<xsd:complexType name="GenericDataAccessCapabilities">
  <xsd:annotation>
    <xsd:documentation>Describes the capabilities of this server.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="SupportedDataObjects" type="tns:ArrayOfSupportedDataObject" />
    <xsd:element name="Properties" type="tns:ArrayOfNameValuePair" />
  </xsd:sequence>
</xsd:complexType>

<!-- GetCapabilities -->
<xsd:element name="GetCapabilities">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for GetCapabilities</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence />
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCapabilitiesResponse">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for response from GetCapabilities</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="getCapabilitiesResult" type="tns:GenericDataAccessCapabilities" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="PutDataResult">
  <xsd:annotation>
    <xsd:documentation>Contains information regarding the response to a DataOperation outside of the returned data itself.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Status" type="tns:PutDataResultStatus" minOccurs="1" />
    <xsd:element name="Id" type="xsd:string" minOccurs="1" />
    <xsd:element name="SuppMsg" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="PutData">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for PutData</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="abs:abstractDataObject" />
      <xsd:element name="options" type="tns:NameValuePair" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="PutDataResponse">
  <xsd:annotation>
    <xsd:documentation>SOAP Message for response from PutData</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="putDataResult" type="tns:PutDataResult" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- Faults -->
<xsd:element name="UnsupportedOptionFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="OptionName" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="UnsupportedDataObjectFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="DataObjectName" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="InvalidXPathFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="XPath" type="xsd:string"/>
            <xsd:element name="XPathProcessorError" type="xsd:string"/>
            <xsd:element name="Position" type="xsd:integer" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="UnknownUidFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Uid" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="UnsupportedCriterionFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="CriterionName" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="MissingParameterFault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Parameter" type="xsd:string" minOccurs="1" maxOccurs="1" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:element>
</xsd:complexType>
</xsd:element>
</xsd:complexType>
</xsd:element>
</xsd:element>
</xsd:complexType>
</xsd:element>
</wsdl:types>

<!-- GetCapabilities -->
<wsdl:message name="GetCapabilitiesSoapIn">
  <wsdl:part name="parameters" element="tns:GetCapabilities" />
</wsdl:message>
<wsdl:message name="GetCapabilitiesSoapOut">
  <wsdl:part name="parameters" element="tns:GetCapabilitiesResponse" />
</wsdl:message>

<!-- PutData -->
<wsdl:message name="PutDataSoapIn">
  <wsdl:part name="parameters" element="tns:PutData" />
</wsdl:message>
<wsdl:message name="PutDataSoapOut">
  <wsdl:part name="parameters" element="tns:PutDataResponse" />
</wsdl:message>

<!-- Faults -->
<wsdl:message name="MissingParameterFaultMessage">
  <wsdl:part name="fault" element="tns:MissingParameterFault" />
</wsdl:message>
<wsdl:message name="UnsupportedOptionFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedOptionFault" />
</wsdl:message>
<wsdl:message name="UnsupportedDataObjectFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedDataObjectFault" />
</wsdl:message>
<wsdl:message name="InvalidXPathFaultMessage">
  <wsdl:part name="fault" element="tns:InvalidXPathFault" />
</wsdl:message>
<wsdl:message name="UnknownUidFaultMessage"
<wsdl:part name="fault" element="tns:UnknownUidFault" />

</wsdl:message>

<wsdl:message name="UnsupportedCriterionFaultMessage">
  <wsdl:part name="fault" element="tns:UnsupportedCriterionFault" />
</wsdl:message>

<wsdl:portType name="PROD_GenericDataAccessSoap">
  <wsdl:operation name="GetCapabilities">
    <wsdl:input message="tns:GetCapabilitiesSoapIn"
    <wsdl:output message="tns:GetCapabilitiesSoapOut"
  </wsdl:operation>
  <wsdl:operation name="PutData">
    <wsdl:input message="tns:PutDataSoapIn"
                wsaw:Action="http://www.prodml.org/api/210/genericDataAccess/PutData" />
    <wsdl:output message="tns:PutDataSoapOut"
                 wsaw:Action="http://www.prodml.org/api/210/genericDataAccess/PutDataResponse" />
    <wsdl:fault name="UnsupportedOptionFault"
                message="tns:UnsupportedOptionFaultMessage" />
    <wsdl:fault name="UnsupportedDataObjectFault"
                message="tns:UnsupportedDataObjectFaultMessage" />
    <wsdl:fault name="UnknownUidFault"
                message="tns:UnknownUidFaultMessage" />
    <wsdl:fault name="MissingParameterFault"
                message="tns:MissingParameterFaultMessage" />
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="PROD_GenericDataAccessSoap"
type="tns:PROD_GenericDataAccessSoap">
  <wsp:PolicyReference URI="#SigEncr"
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <!-- GetCapabilities Operation -->
  <wsdl:operation name="GetCapabilities">
    <soap:operation
      soapAction="http://www.prodml.org/api/210/genericDataAccess/GetCapabilities" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
<wsdl:operation>
  <!-- PutData Operation -->
  <wsdl:operation name="PutData">
    <soap:operation
      soapAction="http://www.prodml.org/api/210/genericDataAccess/PutData" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="UnsupportedOptionFault">
      <soap:fault use="literal" name="UnsupportedOptionFault" />
    </wsdl:fault>
    <wsdl:fault name="UnsupportedDataObjectFault">
      <soap:fault use="literal" name="UnsupportedDataObjectFault" />
    </wsdl:fault>
    <wsdl:fault name="UnknownUidFault">
      <soap:fault use="literal" name="UnknownUidFault" />
    </wsdl:fault>
    <wsdl:fault name="MissingParameterFault">
      <soap:fault use="literal" name="MissingParameterFault" />
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="PROD_GenericDataAccess">
  <wsdl:port name="PROD_GenericDataAccessSoap" binding="tns:PROD_GenericDataAccessSoap">
    <soap:address location="http://127.0.0.1:7521/" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>