SensorNet Application/Service Layer

- **Sensors & Sensor Suites**: measurement updates
- **H&E Models**: alerts, sensor readings, registration, model results
- **User Applications**: sensor readings, alerts, status, model results

**Network Layer**

**Service Layer**

OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE
Application/Service Layer Ideals

- All communication issues handled at lower layers
- No special interface software
  - Custom protocol stacks
  - Special networking libraries
Application/Service Layer Goals

• Common interfaces for
  – Sensors and sensor suites
  – Models
  – User/operator applications

• Hide details and effects of
  – Transport layer networking (and below)
  – Data architecture and its implementation

• Ride the wave of standards
  – Web services
  – SOAP
Approaches

• Suggest a service-oriented “API” for SensorNet
  – Build to interfaces and schema
• SOAP
  – Leverages standardization activities
  – XML Encryption
  – XML Signatures
  – SAML
  – Project Liberty
• Spec-itecture
  – Standards and interfaces
Domain Patterns

- Self service
- Collaboration
- Context aggregators (portal)
- External processing/extended enterprise
State of Web Services

Open XML Description (WSDL)

Open Transport (HTTP, TCP/IP)

Open Messaging (SOAP)
Application/Service Security

- Authentication
- Identity propagation
- Authorization
- Encryption
- Non-repudiation
Web Services

• Ambiguous term with its own range of meanings
• Generally, communication of XML documents across HTTP
• Can mean
  – REST
  – SOAP-RPC
  – Document style SOAP
Communication Approaches

• Range of network and transport layer options
  – Closed, application-specific
    • Socket level exchange of packed binary message content
  – Distributed object middleware, structured
    • CORBA
  – Service-oriented, structured
    • SOAP
  – Service-oriented, unstructured
    • REST
Web Services

• REST architectural style
  – Representational State Transfer
    • http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
    • http://www.prescod.net/rest/
  – Distinguishing characteristics
    • Uniform interface between components
    • Key abstraction is a resource, anything that can be the target of a URI/URL
    • HTTP verbs are enough (GET, PUT, POST, DELETE, etc.)
    • URLs identify resources
    • Web of linked resources
Web Services

• SOAP-RPC
  – Simple Object Access Protocol, Remote Procedure Call
  – Most common interpretation of “Web services”
  – Heavily supported with tools
  – Web Services Description Language (WSDL) to describe service interface and end points
  – Universal Description, Discovery, and Integration (UDDI) services to find Web services
Web Services

- **SOAP-RPC Issues**
  - Rather difficult to deal with complex data
    - Limitations in XML Schema definitions
    - Cannot specify variant (IDL *Any*) fields or parameters
  - Service bound to WSDL definition
    - Just as CORBA servant bound to IDL definition
    - Although WSDL processed at run time, programmer must know interfaces and semantics of the call
    - Tightly coupled, fine grain
  - Paul Prescod’s arguments for REST
    - [http://www.prescod.net/rest/rest_vs_soap_overview/](http://www.prescod.net/rest/rest_vs_soap_overview/)
    - [http://www.sys-con.com/xml/article.cfm?id=454](http://www.sys-con.com/xml/article.cfm?id=454)
Web Services

• Document style SOAP
  – SOAP message envelopes but not RPC
    • Use header for message-level services, like security
  – Compromise between RPC and REST?
  – Loosely coupled, coarse grain
  – Transmit XML document in SOAP body
    • Can specify schema in WSDL “contract”
    • Schema and/or content can change (non-destructively) without changing messaging infrastructure
  – SOAP with attachments (SwA) can include binary attachments
Web Services

• Document style SOAP (cont’d)
  – Can use any transport medium
    • Socket
    • HTTP servlet or CGI
    • Messaging service
  – Example: DocSOAP from CommerceOne
    • http://www.commerceone.com/developers

• Document style in general used by OpenGIS
  Web Services Initiative specifications
Examples

```xml
<snet:request xmlns:tns="http://service.snet.ornl.gov/request"
              xmlns:snet="http://service.snet.ornl.gov/SNet">
  <snet:GetSites spatialDomain="-79,35,5,5"/>
  <!-- SAML assertions ? -->
</snet:request>

<snet:response xmlns:snet="http://service.snet.ornl.gov/SNet"
               xmlns:tns="http://service.snet.ornl.gov/response">
  <timestamp>20030731151700</timestamp>
  <Sites count="12">
    <Site>
      <SiteID>DC005</SiteID>
      <SiteName>Arboretum</SiteName>
      <Latitude>38.9160995</Latitude>
      <Longitude>-76.9688034</Longitude>
      <City>Washington</City>
      <State>DC</State>
    </Site>
    ...
  </Sites>
</snet:response>
```
Examples

<snet:request xmlns:tns="http://service.snet.orl.gov/request"
             xmlns:snet="http://service.snet.orl.gov/SNet">
  <snet:GetSensorReadings
       spatialDomain="-77.24,38.787811,0.378004,0.222189"
       time="200307301800" />
</snet:request>

<snet:response xmlns:snet="http://service.snet.orl.gov/SNet"
               xmlns:tns="http://service.snet.orl.gov/response">
  <timestamp>200307301801</timestamp>
  <SensorReadings count="7">
    <Station>
      <SiteID>DC005</SiteID> <SiteName>Arboretum</SiteName>
      <Date>20030730</Date> <Hours>18.00</Hours>
      <Latitude>38.9160995</Latitude> <Longitude>-76.9688034</Longitude>
      <BattVolts>12.81</BattVolts> <RelHumidity>26.97</RelHumidity>
      <TempBar>1.212</TempBar> <TempMax>1.588</TempMax>
      <TempMin>0.658</TempMin> <TempStdDev>0.161</TempStdDev>
      <WindDir>239.1</WindDir> <WindDirStdDev>24.53</WindDirStdDev>
      <WindSpeed>1.327</WindSpeed> <WindVecMag>1.205</WindVecMag>
    </Station>
    ...
  </SensorReadings>
</snet:response>
Grid Services

• Grid concepts
  – Coordinated resource sharing and problem solving
  – Virtual organizations
  – Computational, data, and access grids

• Globus project toolkit 3
  – Based on Open Grid Services Architecture (OGSA)
  – Implements Open Grid Services Infrastructure (OGSI)
    • “Standard web services” supporting grid computing
  – Built on SOAP-RPC
WS Security Standards

• Examples
  – XML Signature
  – XML Encryption
  – XML Key Management Specification (XKMS)
  – XML Access Control Language (XACL)
  – Security Assertion Markup Language (SAML)
  – Project Liberty

• Can do none of it and rely on TLS/SSL
  – Encrypt everything
WS Security Standards

• Relationships of standards

XKMS and other

Username  XrML  XCBF
SAML  Kerberos  X.509

WSS

XML Signature  XML Encryption

SOAP, SwA
WS Security Standards

• XML Signature
  – Allows digital signature for whole or partial XML document
  – Processes for signature validation
  – <Signature> element added to SOAP header
  – Types
    • Enveloped (signs content containing the signature)
    • Enveloping (signs an object element of the signature itself)
    • Detached (signs content external to the signature)
  – http://www.w3.org/Signature/
• Detached signature example

```xml
<SignedInfo>
  <CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsash1"/>
  <Reference URI="http://some.where.org/content/">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>j6lx3rvEPO0vKtMup4NbeVu8nk="</DigestValue>
  </Reference>
</SignedInfo>
<SignedInfo>
  <SignatureValue>MC0CFFrVtRlk="</SignatureValue>
</SignedInfo>
</Signature>
```
WS Security Standards

• XML Encryption
  – Encrypting or decrypting whole or partial XML document or binary data
  – References XML Signature specification
  – `<EncryptedData>` element(s) in SOAP body (and header)
  – Can encrypt leaf element or element tree

WS Security Standards

- XML Encryption example

```xml
<PaymentInfo xmlns="http://example.org/paymentv2">
    <Name>John Smith</Name>
    <CreditCard Limit="5,000" Currency="USD">
        <Number>4019 2445 0277 5567</Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>04/02</Expiration>
    </CreditCard>
</PaymentInfo>

<PaymentInfo xmlns="http://example.org/paymentv2">
    <Name>John Smith</Name>
    <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
        xmlns="http://www.w3.org/2001/04/xmlenc#">
        <CipherData>
            <CipherValue>A23B45C56</CipherValue>
        </CipherData>
    </EncryptedData>
</PaymentInfo>
```
WS Security Standards

- **SAML**
  - XML framework for exchanging authentication and authorization information
  - Assertions handed out by “SAML authorities”, themselves web services

  - [http://www.opensaml.org](http://www.opensaml.org)
**WS Security Standards**

- **SAML scenario**

1. Client requests service, service redirects
2. Client authenticates, receives assertions
3. Client resends with assertions
4. Service verifies assertions
5. Service response

**Diagram:**
- **Client**
- **Service**
- **SAML Authority**
WS Security Standards

• Project Liberty
  – Uses SAML for single sign-on (SSO)
  – Identity Federation Framework (ID-FF)
    • Identity/account linkage, SSO, session management
  – Identity Service Interface Specification (ID-SIS)
    • Schema and specs for interoperable identity services
  – Identity Web Services Framework (ID-WSF)
    • Framework for identity-based web services
    • Includes SOAP binding
    • Dynamic discovery of identity services
    • SOAP Authorization Service to authenticate parties communicating via SOAP

  – http://www.projectliberty.org
Other Technologies

• JXTA
  – Peer-to-peer (P2P) discovery and virtual networking
  – Rendezvous peers can forward between firewalls and NAT addressing over HTTP (if firewall allows)
  – Collection of protocols
    • peer discovery, peer information, peer resolver, pipe binding, endpoint routing, rendezvous

  – http://www.jxta.org
Other Technologies

- Jini
  - Adaptive, resilient, network-centric services
  - Based on Java, requires Java proxies
  - Code mobility
  - JavaSpaces “virtual space” Jini service

- [http://www.jini.org](http://www.jini.org)
Specifications Related to SensorNet

- OpenGIS SensorWeb
  - Sensor Modeling Language (SensorML)
  - Sensor Collection Service
    - http://ip.opengis.org/swe