Advanced SensorNet Technologies to Monitor Trusted Corridors: SensorNets for Multiple Owners

a Transportation Security SensorNet (TSSN)

Gary J. Minden

Electrical Engineering & Computer Science

Information & Telecommunication Technology Center
KU SensorNet Team

**Faculty**
- Victor S. Frost
- Dan Deavors
- Gary J. Minden
- Joseph B. Evans

**Research Staff**
- Ed Komp
- Leon Searl

**Students**
- Adam Powell
- Martin Kuehnhausen
- James Stevens
- Daniel Fokum
- Angela Oguna
- Matthew Zeets
- George Cox
Outline

• Tasks and Objective
• Rail Technology Proof of Concept Field Trial and Plans
• General Framework for Transportation Security SensorNet (TSSN)
• Specifics on TSSN Implementation
• Status
• Demonstration of Initial Capabilities
Tasks and Objectives

• Rail Technology Proof of Concept and Integration of the SmartPort Trade Data Exchange (TDE) Architecture and Transportation Security SensorNet (TSSN) Technologies

• Access/control/security mechanisms to support the Transportation Security SensorNet (TSSN)

• System modeling, measurement, and performance

• Communications technologies for the continuous monitoring of containers and communications to the central SmartPort data clearinghouse

• RFID applications in the trusted corridor
Rail Technology Proof of Concept

Additional Sensor  
Vehicle Mounted Seal/Tag Reader/Writer  
Vehicle Mounted SensorNet Collector  
Satellite/Cellular Communications  
IP Internet  
SOA  
TDE Applications  
Other Applications  
IP Internet  
Trade Data Exchange  
Command & Control  
Other Users
Rail Technology Proof of Concept

Container

Intelligent RFID Enabled Sensor Access Point

Intelligent RFID Enabled Cable Seal

Carrier Shipment Information

Network Control Center

Trade Data Exchange

Operational Test Environment
General Framework for Transportation Security SensorNet (TSSN)

- Functional description of resources
- Map SensorNet functions onto hardware and software
- Comprehensive network infrastructure
- Multiple owners and operators of infrastructure
General Framework for Transportation Security SensorNet (TSSN)
Approach for TSSN

- Move to Service Oriented Architecture (SOA)
- Modify current Ambient Computing Environment (ACE) to use SOA
  - Standards based
  - Improved interoperability with other platforms (C#, .Net, …)
  - Remove Java dependency
- Service and clients specific to Transportation Security Sensor Net (TSSN) built on top of ACE/SOA
- The result is an ACE/SOA Infrastructure
ACE/SOA Infrastructure

• Infrastructure based Service Oriented Architecture
• One or more enterprises of clients utilizing services
• Clients utilizing services can be Intra- and Inter-Enterprise
• Infrastructure is based on W3C and OASIS standard Web Service specifications
  • Service discovery utilizes Universal Description Discovery and Integration v3 (UDDI)
  • Services are described using Web Service Description Language (WSDL)
  • Client/Server Communication based on Simple Object Access Protocol v1.2 (SOAP)
    – Message Identification and Routing handled by WS-Address
    – Message Data Confidentiality handled by WS-SecureConversation
    – Message Authentication handled by WS-Security
    – Message Authorization handled by WS-Authorization (when written)
    – Event Notification handled by WS-BasicNotification
• Clients from Federation A are allowed to use service from Federation B using WS-Trust and WS-Federation specifications
ACESOA Infrastructure Services

- **Service Directory**
  - Clients query service information from Service Directory (UDDI)
  - Service Information includes (via WSDL)
    - Network location
    - Security Policies (Authorization, Authentication, Privacy)
    - Service Interface Information
      - Procedures and arguments
      - Service Resources (example: Sensor Information via SensorML added to WSDL)

- **Authentication**
  - Clients obtain valid authentication tokens from this service
  - Services validate authentication tokens with this service

- **Authorization**
  - Services obtain authorization certificates from this service
ACESOA Infrastructure Implementation

• Existing Implementation
  • SOAP, WSDL ↔ Apache Axis2, gSOAP (for embedded)
  • WS-Security, WS-SecureConversation ↔ Apache Rampart
  • WS-BaseNotification ↔ Apache Muse
  • UDDI Server V3 ↔ OpenUDDI
  • UDDI Browser ↔ Novell UDDI Browser

• ITTC Implementation
  • Server Publication to UDDI
  • Authentication Server
  • Accessing Authentication Tokens by clients servers
  • Authorization Server
  • Accessing and Verifying Authorization certificates by servers
    – WS-Authorization

• With usage of accepted OASIS and W3C specifications many parts of implementation can be replaced with parts from other open sources
ACESOA Messaging Layers

Client

ACE Infra

Soap
Serialze
XML
WS-Authorization
Authentication
WS-Security (Encryption)
WS-Addressing (ID, Routing)
Transport (HTTP, SMTP, TCP)

Server

ACE Infra

Soap
Serialze
XML
WS-Authorization
Authentication
WS-Security (Encryption)
WS-Addressing (ID, Routing)
Transport (HTTP, SMTP, TCP)

Author Cert

Authen Token
ACESOA Federation of Services Overview

Enterprise A
- Client
- Authenticate
- Service Dir

Enterprise B
- Service Dir
- Authenticate
- Authorize
- Sense
- Proxy
- Service Dir
- Remote Authenticate
- Authorize
- Sensor
- Sense Proxy
- Sense Filter
- Sense DB
ACESOA Remote Service Messaging

- Authenticator Service – provides and verifies Identity Tokens
- Remote Sensor Service – on a railcar
- Proxy Sensor Service – on enterprise network and connected to modem
- Obtain Sensor Data
  - Obtain Remote Sensor Service Info
    - Setup by operator manually
    - Includes Service GSM Phone #
  - Obtain Authentication Token
  - Call Sensor
    - Establish GSM link
  - Get Sensor Data
    - getData
      - Includes Authent Token
    - Check Authentication Token against file
      - Local file used to save GSM traffic
    - Check Authorization against certificate in file
      - Local file used to save GSM traffic
    - Grab data from log/or current reading
    - RtnData
**ACESOA Remote Event Notification**

- Sensor Proxy has established connection to Remote Sensor and authenticated
  - Authentication and Authorization on Remote Sensor
- Sensor Proxy subscribes to specific Sensor event/alarm.
  - Authentication and Authorization on Remote Sensor
- Sensor Proxy disconnects from Remote Sensor
  - Drops GSM connection
- Event/Alarm occurs on sensor
- Sensor Notifies Event Subscriber (Sensor Proxy)
  - Sensor obtains Sensor Proxy GSM PH#
    - Sensor establishes GSM connection
    - Sensor obtains Authentication Token
    - Sensor sends Event Notification
    - Sensor Proxy checks authentication
    - Sensor Proxy verifies authorization
    - Sensor Proxy drops GSM connection
Trust established by humans
Service Directories subscribe to public service publications
  - Sensor DB make public publication
Client (EntB) wants to get data from SensorDB (EntA)
  - Ask Service Directory for SensorDB information
  - Get authentication token
  - Get data from Sensor DB
TSSN Implementation Overview

TDE

VNOC
- SMS Alarm
- Display Alarm

Sat. Provider

MRN
- Wireless Seal & Tag Reader/Writer
- SensorNet Collector
- GPS Receiver
- Rail Car

Internet

ShipementData
- Train/Sensor Ids
- Alarms
- Location

Sensor Measurements
- Sensor Alarms
- Sensor Config
- Location
TSSN Virtual Network Operations Center

- **UDDI Service Directory**
- **Authentication**
- **Authorization**

Communications links not shown

- **Sensor Alerting/ Web Notification Service**
  - SensorML
  - SensorConfig

- **Event Response & Rules**
  - SensorAlarm/ShipData

- **Train Subscr/Sn_train_data**

- **Sensor Search/Resp**
  - Catalog Service

- **Sensor Management**

- **Mobile Rail Net (MRN)**
  - Meas Req/Resp
  - SetAlarmRules
  - Alarm Subscr/Alarm Pub.

- **Sensor Observation Service**

- **Sensor Data**

- **Event Data**
  - SMTP/SMS

- **Rule Data**

- **Web Server**

- **User Alarm Client**

- **Web Feature Service**

- **TrainPosReq/Resp**
TSSN Mobile Rail Network

Communications links not shown

UDDI Service Directory
Authentication
Authorization

Catalog Service
Sensor Discovery
GPS Reader
EdgeNode Sensor Measurement DB
GPS Device

Sensor Observation Service
Sensor ML

Sensor Alerting Service
SensorML
Alarmsubscr/Event
SetAlarmRules
Alarm Subscr/Alert Pub.

User Alarm Client

Event DB
Sensor Configuration
Meas Req/Resp

Sensor Alerting Service

Web Feature Service

Lat/Lon Req/Resp

ORNL Implementation
TSSN Implementation
ACE Implementation
OGC Implementation/Interface

High-G Tech Tags/Seals

Sense Meas DB
TSSN Implementation

- Sensor, Filter and Database services
  - May use hierarchy of services to enable code reuse
- Clients for TSSN Services
- Clients and Services on PCs written in Java
- Services on embedded devices written in C
Status

• **ACE Service Oriented Architecture (ACE SOA)**
  • SOA Infrastructure for TSSN
• **Transportation Security SensorNet (TSSN)**
  • Phase1 – Simple service messages based on OGC specifications (used in trials)
  • Phase2 – Use full OGC specification interfaces messages.
  • Phase3 – Use lessons learned from Phase1 and 2 to make improvements
# Status ACE SOA

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport</strong></td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- SOAP/HTTP</td>
</tr>
<tr>
<td></td>
<td>- Apache Axis2 implementation (v1.4)</td>
</tr>
<tr>
<td><strong>Confidential Data</strong></td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- WS-Security, HTTPS</td>
</tr>
<tr>
<td></td>
<td>- Axis2 implementation (Rampart)</td>
</tr>
<tr>
<td><strong>Remote Exception</strong></td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- SOAP Fault</td>
</tr>
<tr>
<td></td>
<td>- Axis2 has extendable exception mechanism</td>
</tr>
<tr>
<td><strong>Server Alert/Alarm</strong></td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- WS-Eventing</td>
</tr>
<tr>
<td></td>
<td>- Implemented mechanism for stand alone clients to receive events.</td>
</tr>
<tr>
<td></td>
<td>- TODO: Switch over to WS-BasicNotification</td>
</tr>
</tbody>
</table>
# Status ACE SOA

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication, Client-Server</td>
<td>100%</td>
</tr>
<tr>
<td>- WS-Security</td>
<td></td>
</tr>
<tr>
<td>- Axis2 implementation (Rampart)</td>
<td></td>
</tr>
<tr>
<td>Intra-Enterprise Authentication Service (signed token)</td>
<td>80%</td>
</tr>
<tr>
<td>- Axis2 implementation (Rampart)</td>
<td></td>
</tr>
<tr>
<td>- Use Standard Token Service (STS)</td>
<td></td>
</tr>
<tr>
<td>- TODO: decide on token and signing mechanism/type</td>
<td></td>
</tr>
<tr>
<td>- Decision may be impacted by load data rate of Iridium</td>
<td></td>
</tr>
<tr>
<td>Inter-Enterprise Authentication Trust</td>
<td>20% Complete</td>
</tr>
<tr>
<td>- WS-Federation/WS-Trust specifies mechanism (Axis2 Rampart)</td>
<td></td>
</tr>
<tr>
<td>- Use Standard Token Service (STS)</td>
<td></td>
</tr>
<tr>
<td>- TODO: Verify that implementation provides required functionality</td>
<td></td>
</tr>
<tr>
<td>Intra-Enterprise Authorization Service</td>
<td>0%</td>
</tr>
<tr>
<td>- WS-Authorization has not been written</td>
<td></td>
</tr>
<tr>
<td>- Can use xACML as language</td>
<td></td>
</tr>
</tbody>
</table>
# Status ACE SOA

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TDE Integration</strong></td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>- Testing with local TDE using EDS provided WSDL</td>
</tr>
<tr>
<td></td>
<td>- TODO: use EDS located TDE service.</td>
</tr>
<tr>
<td><strong>Intra-Enterprise Service Discovery</strong></td>
<td>90%</td>
</tr>
<tr>
<td></td>
<td>- UDDI v3, OpenUDDI Implementation (v0.9.8)</td>
</tr>
<tr>
<td></td>
<td>- Implemented common service code for automatic publishing</td>
</tr>
<tr>
<td></td>
<td>- Implemented common client code for simplified service discovery.</td>
</tr>
<tr>
<td></td>
<td>- TODO: complete enterprise service to clean up stale UDDI info.</td>
</tr>
<tr>
<td><strong>Inter-Enterprise Service Discovery</strong></td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>- UDDI v3, OpenUDDI Implementation replication</td>
</tr>
<tr>
<td></td>
<td>- Each enterprise has public UDDI for replication with other enterprise public UDDIs</td>
</tr>
<tr>
<td></td>
<td>- TODO: Need enterprise service to publish public services to public UDDI.</td>
</tr>
</tbody>
</table>
## Status ACE SOA

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auditing/Monitoring</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>- Implemented message logging module for Axis2</td>
</tr>
<tr>
<td></td>
<td>- Implemented GUI for message monitoring</td>
</tr>
<tr>
<td></td>
<td>- TODO: Evaluate current utility of module and GUI and make improvements.</td>
</tr>
</tbody>
</table>
## Status TSSN Phase 1

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRN Communication Service</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>- Basics of setting up network connection over Iridium and GSM complete</td>
</tr>
<tr>
<td></td>
<td>- Can measure GSM signal strength for connection switch over decision</td>
</tr>
<tr>
<td></td>
<td>- TODO: write SOA service code</td>
</tr>
<tr>
<td>MRN Sensor Node Service</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>- Can processing all service operations (start, stop, GetCapabilities, etc)</td>
</tr>
<tr>
<td></td>
<td>- Can generate alert events based on simulated sensor events</td>
</tr>
<tr>
<td></td>
<td>- TODO: complete last HGT AVL Reader commands</td>
</tr>
<tr>
<td>MRN Alarm Processing Service</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>- Can receive Alerts from Sensor Node</td>
</tr>
<tr>
<td></td>
<td>- Can do simple if/then/else event processing and publish alarms to subscribers</td>
</tr>
<tr>
<td></td>
<td>- TODO finish Complex Event Processing code using Esper.</td>
</tr>
</tbody>
</table>
## Status TSSN Phase 1

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRN Alarm Reporting Client</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- Subscribes to MRN Alarm Processing Service for Alarms</td>
</tr>
<tr>
<td></td>
<td>- Uses simple text output.</td>
</tr>
<tr>
<td>NOC Sensor Management Service</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- Accepts TDE start operation</td>
</tr>
<tr>
<td></td>
<td>- Sends start operation to MRN Sensor Node</td>
</tr>
<tr>
<td>NOC Alarm Processor Service</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>- Subscribes to MRN Alarm Processor for Alarm events</td>
</tr>
<tr>
<td></td>
<td>- Can receive Alarms from MRN Alarm Processor</td>
</tr>
<tr>
<td></td>
<td>- Can do simple if/then/else event processing and publish alarms to subscribers</td>
</tr>
<tr>
<td></td>
<td>- TODO finish Complex Event Processing code using Esper.</td>
</tr>
<tr>
<td>NOC Alarm Reporting Service</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>- Subscribes to NOC Alarm Processor for Alarm Events</td>
</tr>
<tr>
<td></td>
<td>- Receives Alarm events and notifies users by SMS message and/or Email based in information in user notification database</td>
</tr>
</tbody>
</table>
# Status TSSN Phase 2

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRN Alarm Reporting Client</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Change to using a web browser interface</td>
</tr>
<tr>
<td>MRN Sensor Node Service</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Use full OGC service interface (SOS, SAS, FaultReport)</td>
</tr>
<tr>
<td>MRN Alarm Processing Service</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Use full OGC SAS Alert and FaultReport</td>
</tr>
<tr>
<td></td>
<td>- TODO: Develop rules for Complex Event Processing based on GPS, Cargo Info and Sensor Readings</td>
</tr>
<tr>
<td>NOC Sensor Management Service</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Use full OGC FaultReport</td>
</tr>
<tr>
<td>NOC Alarm Processing Service</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Use full OGC FaultReport</td>
</tr>
<tr>
<td></td>
<td>- TODO: Develop rules for Complex Event Processing Develop rules for Complex Event Processing</td>
</tr>
</tbody>
</table>
## Status TSSN Phase 2

<table>
<thead>
<tr>
<th>Functionality</th>
<th>% Complete, Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC Alarm Reporting Service</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Publish user notifications to subscribed clients. This is really just for</td>
</tr>
<tr>
<td></td>
<td>the NOC Alarm Reporting Client</td>
</tr>
<tr>
<td>NOC Alarm Reporting Client</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>- TODO: Similar to MRN Alarm Reporting Client but also subscribes to NOC Alarm</td>
</tr>
<tr>
<td></td>
<td>Reporting Service</td>
</tr>
</tbody>
</table>
Demonstration of Initial Capabilities
Project time line and completion date

Field Trials Completed

Project Complete

With Internal Reallocation

Aug 07  Aug 08  Aug 09  Aug 09

EFM/CTIP/SmartPort Activities
Interim Report
RFID applications
Communications Systems
Data integration and processing
Modeling-systems engineering capability
Intermodal Technology Proof of Concept and Integration
Project time line and completion date

Project Timeline

- Final Report
- RFID applications
- Communications Systems
- Data integration and processing
- Modeling-systems engineering capability
- Intermodal Technology Proof of Concept and Integration

Aug 07 Aug 08 Aug 09

Field Trials Completed
Project Complete
Information and Telecommunications Technology Center