2007

Sensor Networks Survey

Cory Andrew Henson
Wright State University - Main Campus

Satya S. Sahoo
Wright State University - Main Campus

KNO.E.SIS

COLLECTING THE DOTS | CONNECTING THE DOTS

Survey of current

Sensor Network Data Management Frameworks

Outline of Presentation

Two current approaches to sensor data management:

1. Application approach
   - GSN - Global Sensor Networks (DERI)
   - Open-source software framework to manage sensor data
   - http://gsn.sourceforge.net
2. Standards approach
   - SWE – Sensor Web Enablement (OGC)
   - Community-standard language framework to manage sensor data
   - http://www.opengeospatial.org/projects/groups/sensorweb

Application Approaches

1. GSN
   - Global Sensor Network
   - Digital Enterprise Research Institute (DERI)
   - http://gsn.sourceforge.net/

2. Hourglass
   - An Infrastructure for Connecting Sensor Networks and Applications
   - Harvard
   - http://www.eecs.harvard.edu/~syrah/hourglass/

3. IrisNet
   - Internet-Scale Resource-Intensive Sensor Network Service
   - Intel & Carnegie Mellon University
   - http://www.intel-iris.net/

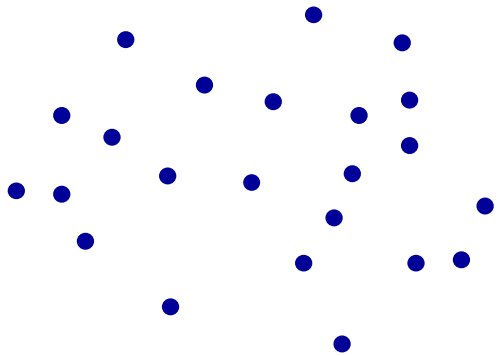
4. SNSP
   - Sensor Network Services Platform
   - University of California, Berkley & DoCoMo
   - http://chess.eecs.berkeley.edu/

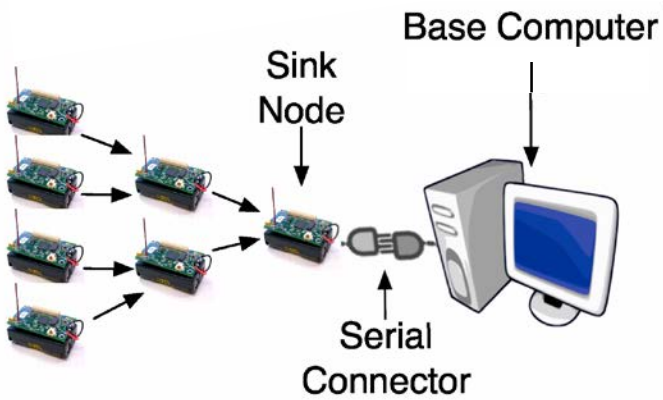
Global Sensor Networks

(GSN)

Middleware for Sensor Networks

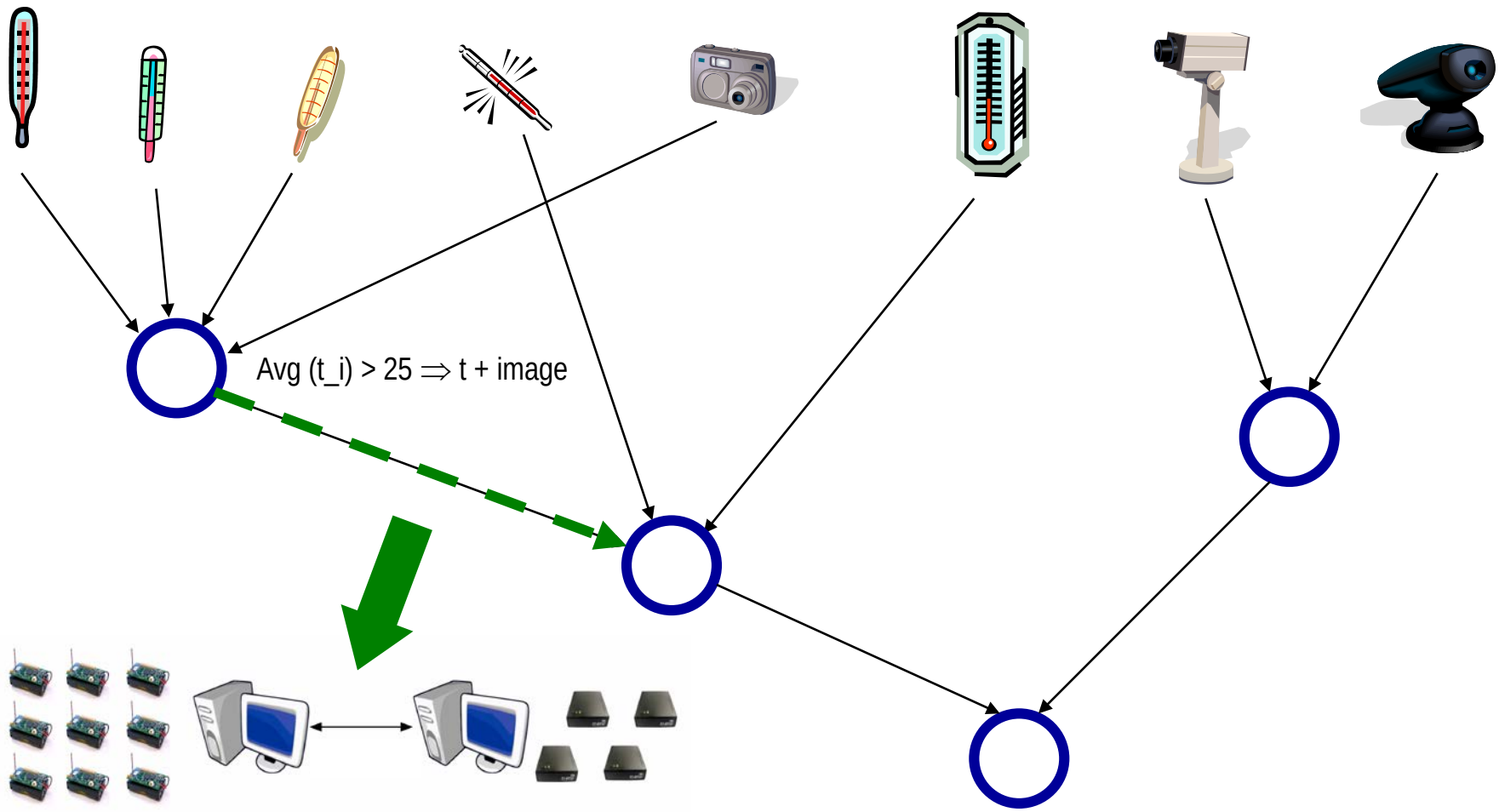
The “Sensor Internet” Vision





- Sensor network + base computer = **sensor node**
- Many sensor nodes produce a lot of data on the Internet
- Questions:
  - Deployment
  - Description
  - Discovery
  - Integration
  - Distributed processing

The “Sensor Internet” at Work



Obstacles on the road to the "Sensor Internet"

Lack of standardization

- High development costs (heterogeneity, novel technologies)
- Limited portability
- No standard APIs and services for fast and flexible development

Dropping prices for sensors technology

⇒ Increasing number of sensor networks and applications

⇒ Large-scale integration of sensor network data

Solutions?

Middleware!

Global Sensor Networks: Design Goals

- **Simplicity**
  - minimal set of powerful abstractions
  - easy to adopt and combine
  - declarative specification of sensor networks and data streams
  - SQL-based query processing
- **Adaptivity**
  - low effort to add new types of sensor networks
  - dynamic (re-) configuration during run-time
- **Scalability**
  - large numbers of data producers and consumers
  - distributed query processing
  - distributed discovery of sensor networks
  - peer-to-peer architecture
- **Light-weight implementation**
  - no excessive hardware requirements
  - standard network connectivity
  - portable (Java-based implementation)
  - easy-to-use, web-based management tools.

Central abstraction: Virtual Sensors

- A virtual sensor can be **any kind of data producer**
  - a real sensor, a wireless camera, a desktop computer, etc.

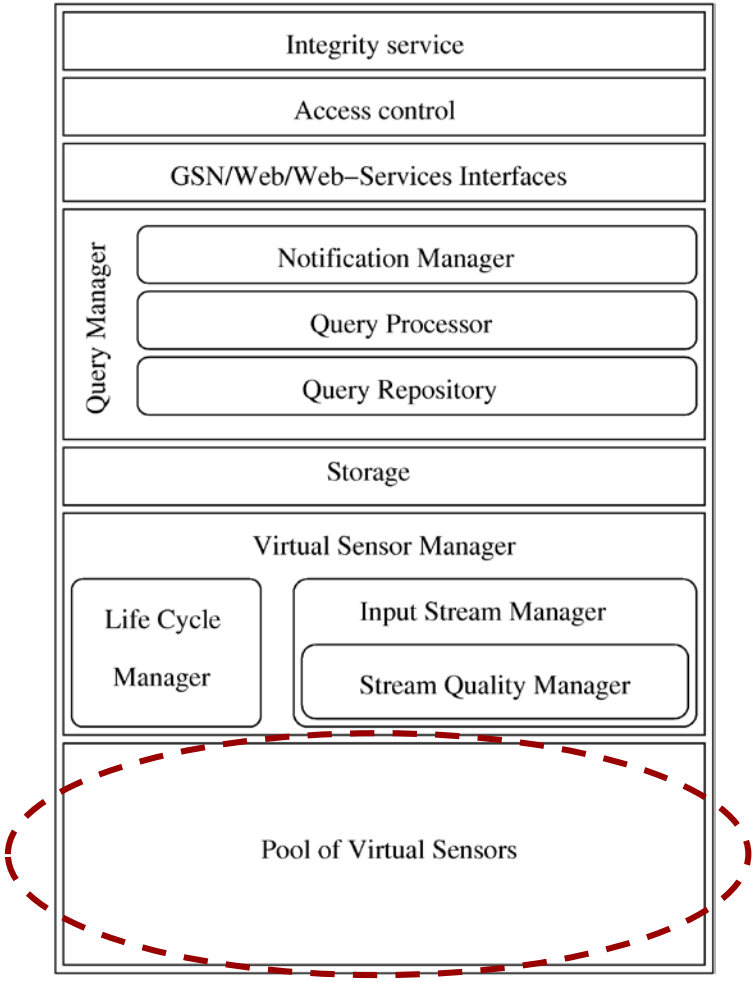
- **Abstract from implementation details**
  - physical sensors
  - a combination of other virtual sensors
  - 1 virtual sensor = *n* input data streams + processing + 1 output data stream

- Specification
  - **metadata** (identification, data, type, location)
  - **structure** and properties of input and output streams
  - **declarative** SQL-based specification of the data stream processing
  - **functional properties** related to stream quality management, persistency, error handling, life-cycle management, and physical deployment.

… oh, it's XML + SQL

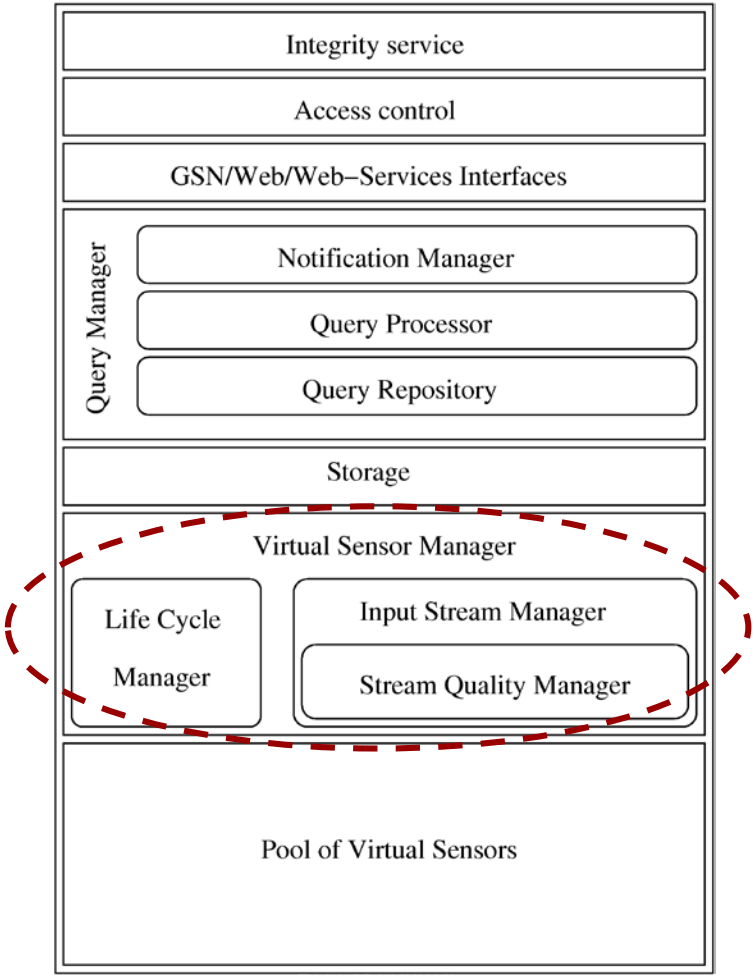
```
<virtual-sensor name="room-monitor" priority="11">
  <addressing>
    <predicate key="geographical">BC143</predicate>
    <predicate key="usage">room monitoring</predicate>
  </addressing>
  <life-cycle pool-size="10" />
  <output-structure>
    <field name="image" type="binary:jpeg" />
    <field name="temp"  type="int" />
  </output-structure>
  <storage permanent="true" history-size="10h" />
  <input-streams>
    <input-stream name="cam">
      <stream-source alias="cam"  storage-size="1"
                     disconnect-buffer-size="10">
        <address wrapper="remote">
          <predicate key="geographical">BC143</predicate>
          <predicate key="type">Camera</predicate>
        </address>
        <query>select * from WRAPPER</query>
      </stream-source>
      <stream-source alias="temperature1“
                     storage-size="1m"
                     disconnect-buffer-size="10">
        <address wrapper="remote">
          <predicate key="type">temperature</predicate>
          <predicate key="geographical">
              BC143-N
          </predicate>
        </address>
        <query>
            select AVG(temp1) as T1 from WRAPPER
        </query>
      </stream-source>
      <stream-source alias="temperature2“
                     storage-size="1m"
                     disconnect-buffer-size="10">
        <address wrapper="remote">
          <predicate key="type">
              temperature
          </predicate>
          <predicate key="geographical">
              BC143-S
          </predicate>
        </address>
        <query>
            select AVG(temp2) as T2
            from WRAPPER
        </query>
      </stream-source>
      <query>
        select cam.picture as image, temperature.T1
                as temp
        from   cam, temperature1
        where  temperature1.T1 > 30 AND
               temperature1.T1 = temperature2.T2
      </query>
    </input-stream>
  </input-streams>
</virtual-sensor>
```

GSN node architecture



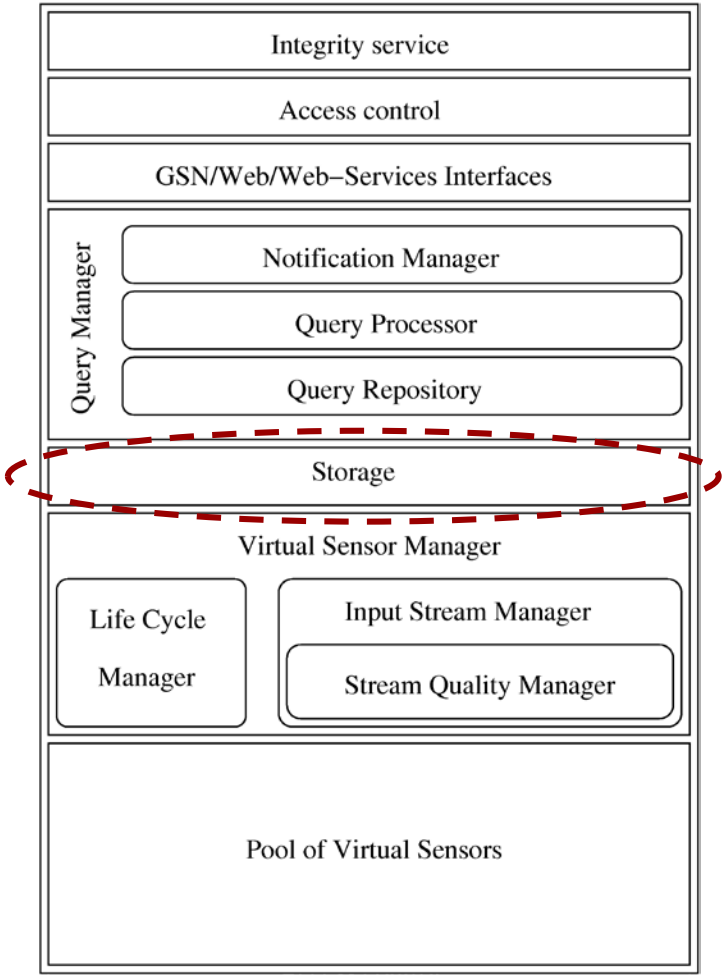
- **Each GSN node hosts a number of virtual sensors**

GSN node architecture



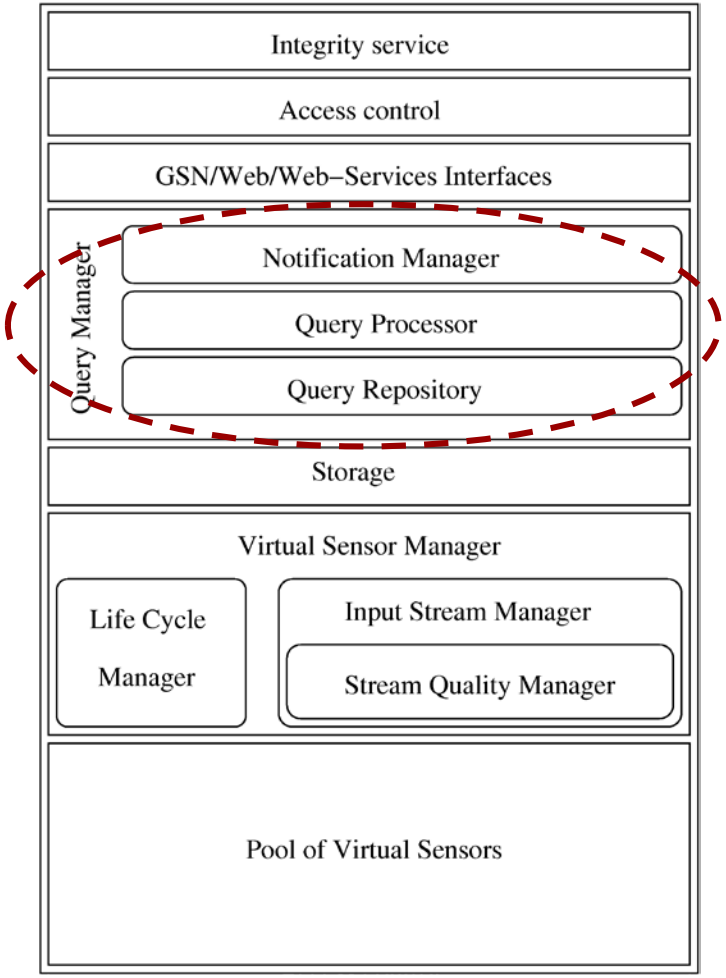
- **Each GSN node hosts a number of virtual sensors**
- **Virtual sensor manager**
  - provides access to the virtual sensors
  - manages the delivery of sensor data
- **Life-cycle manager**
  - provides and manages the resources provided to a virtual sensor
  - manages the interactions with a virtual sensor
  - ensures stream quality
  - manages the life-cycle of sensors

GSN node architecture



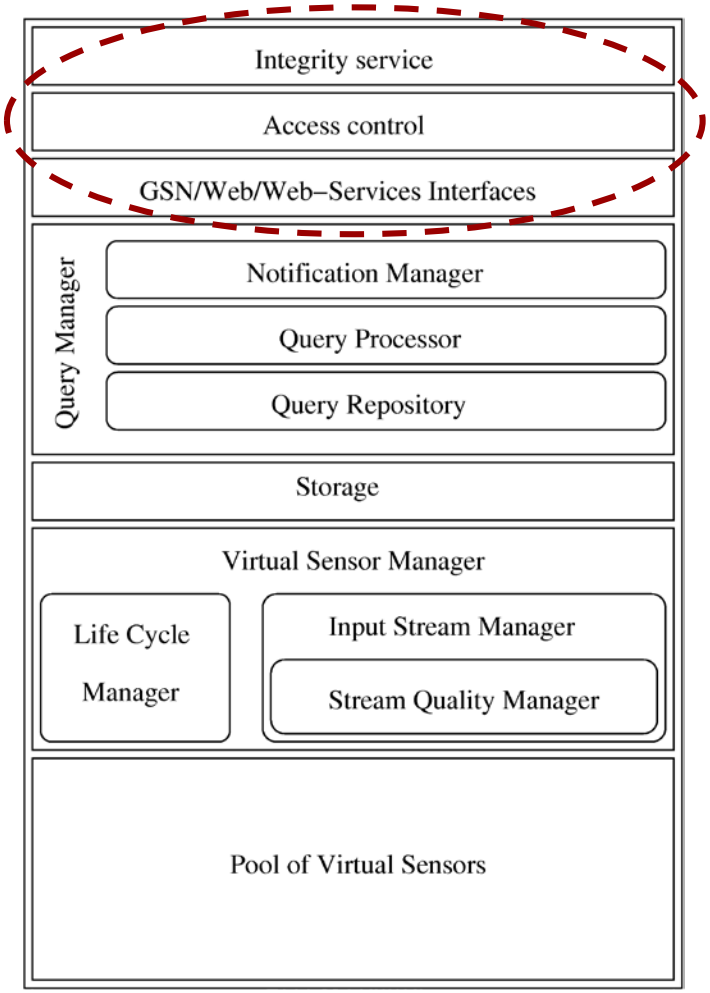
- **Each GSN node hosts a number of virtual sensors**
- **Virtual sensor manager**
  - provides access to the virtual sensors
  - manages the delivery of sensor data
- **Life-cycle manager**
  - provides and manages the resources provided to a virtual sensor
  - manages the interactions with a virtual sensor
  - ensures stream quality
  - manages the life-cycle of sensors
- **Storage layer**
  - persistent storage for data streams

GSN node architecture



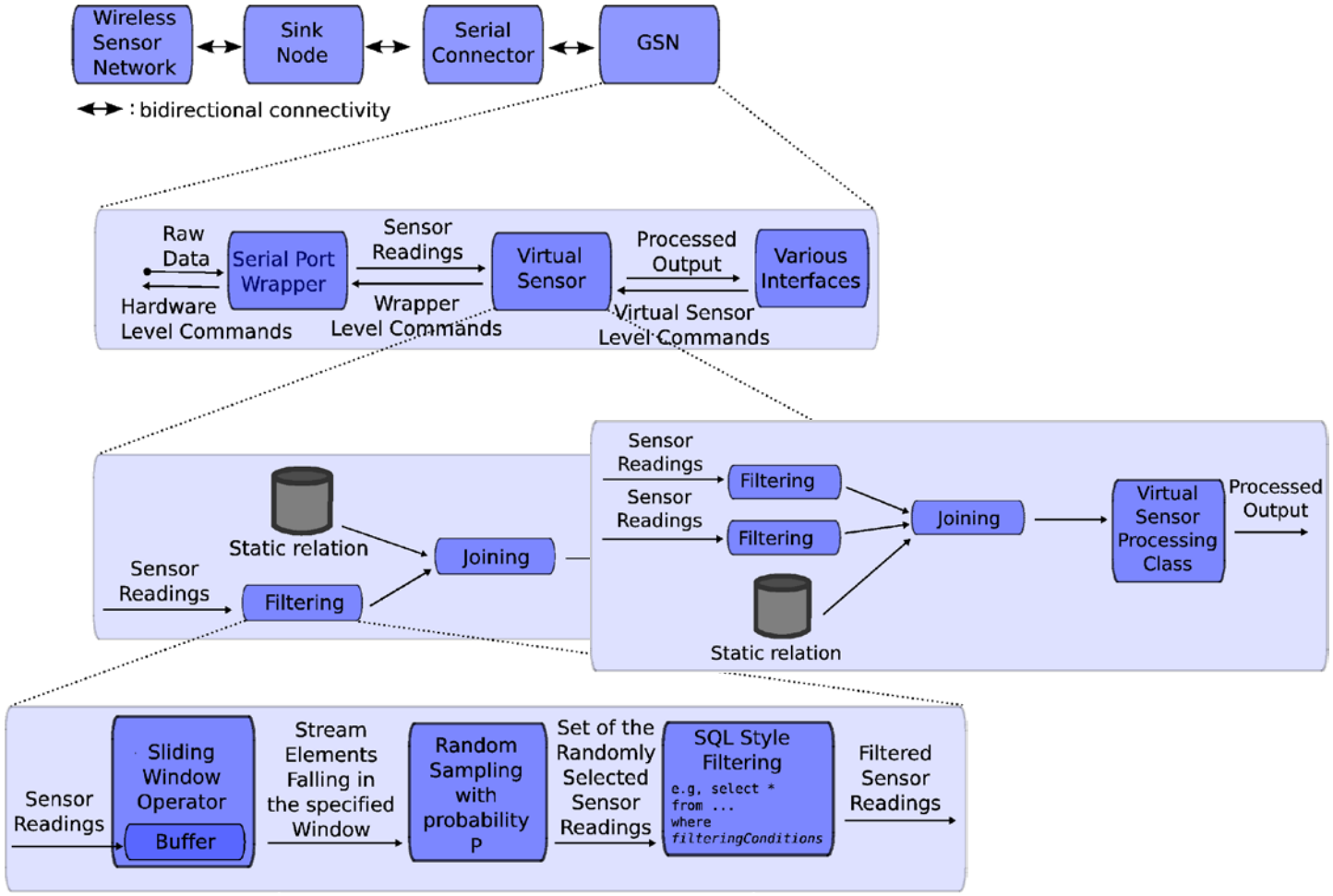
- **Each GSN node hosts a number of virtual sensors**
- **Virtual sensor manager**
  - provides access to the virtual sensors
  - manages the delivery of sensor data
- **Life-cycle manager**
  - provides and manages the resources provided to a virtual sensor
  - manages the interactions with a virtual sensor
  - ensures stream quality
  - manages the life-cycle of sensors
- **Storage layer**
  - persistent storage for data streams
- **Query manager**
  - manages active queries
  - query processing
  - delivery of events and query results to registered, local or remote consumers

GSN node architecture



- **Each GSN node hosts a number of virtual sensors**
- **Virtual sensor manager**
  - provides access to the virtual sensors
  - manages the delivery of sensor data
- **Life-cycle manager**
  - provides and manages the resources provided to a virtual sensor
  - manages the interactions with a virtual sensor
  - ensures stream quality
  - manages the life-cycle of sensors
- **Storage layer**
  - persistent storage for data streams
- **Query manager**
  - manages active queries
  - query processing
  - delivery of events and query results to registered, local or remote consumers
- Top layers: **access**, **access control**, and **integrity**

Local sensor data processing



Accessing physical sensors: Wrappers

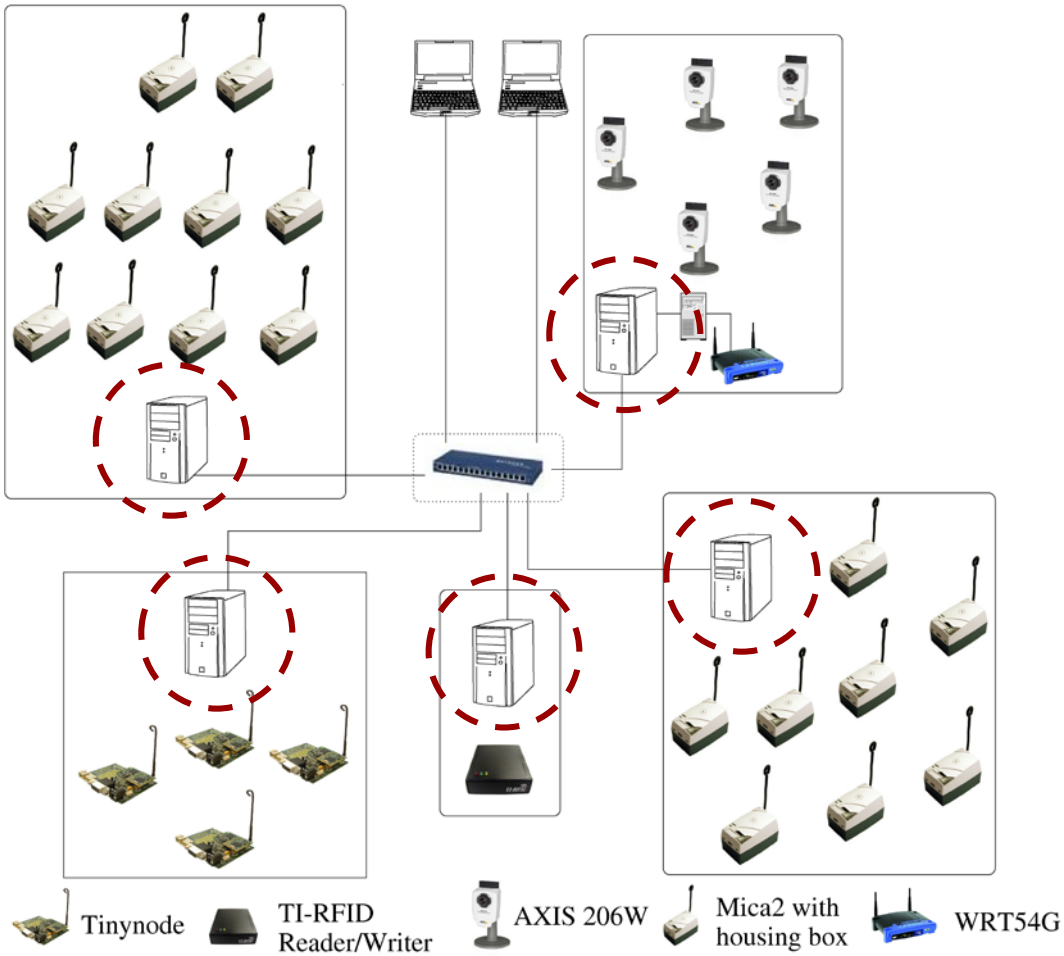
- **HTTP generic wrapper**
  - devices accessible via HTTP GET or POST requests, e.g., the AXIS206W wireless camera
- **Serial forwarder wrapper**
  - enables interaction with TinyOS compatible motes (standard access in TinyOS)
- **USB camera wrapper**
  - Local USB connection
  - supports cameras with OV518 and OV511 chips
- **TI-RFID wrapper**
  - access to Texas Instruments Series 6000 S6700 multi-protocol RFID readers
- **WiseNode wrapper**
  - access to WiseNode sensors (CSEM, Switzerland, http://www.csem.ch/)
- **Generic UDP wrapper**
  - any device using the UDP protocol
- **Generic serial wrapper**
  - supports sensing devices which send data through the serial port

Coding efforts for wrappers

Wrapper type	Lines of code
TinyOS	120
WiseNode	75
Generic UPD	45
Generic serial	180
Wired camera	300
Wireless camera (HTTP)	60
RFID reader (TI)	50

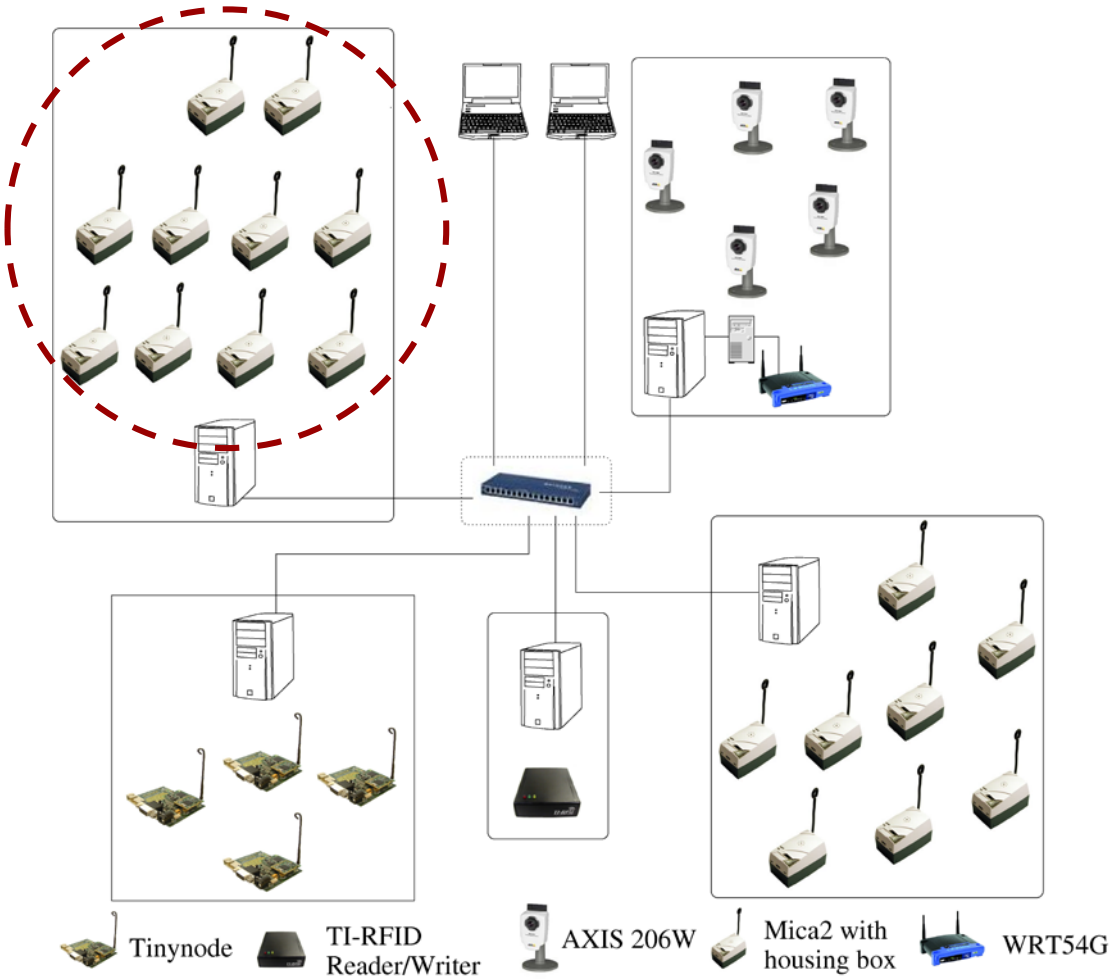
Does it really work?

Experimental setup



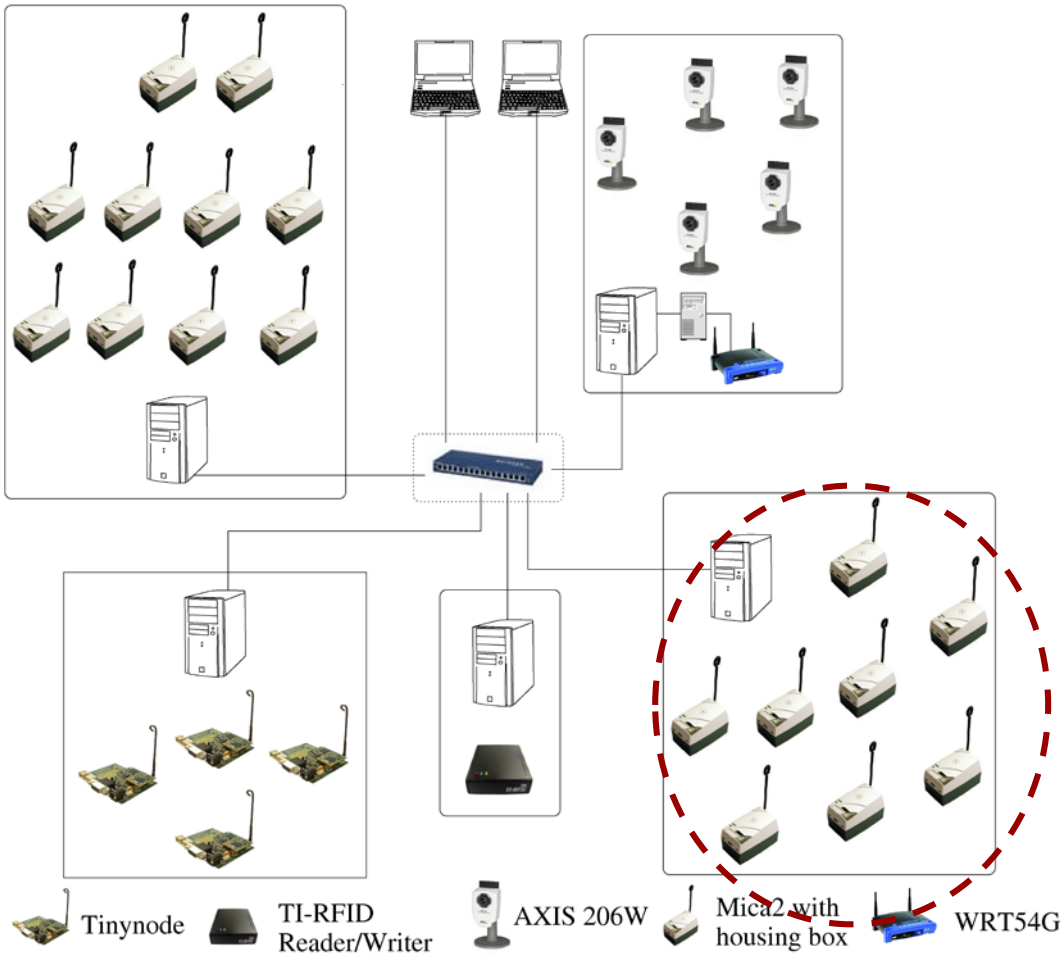
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18

Experimental setup



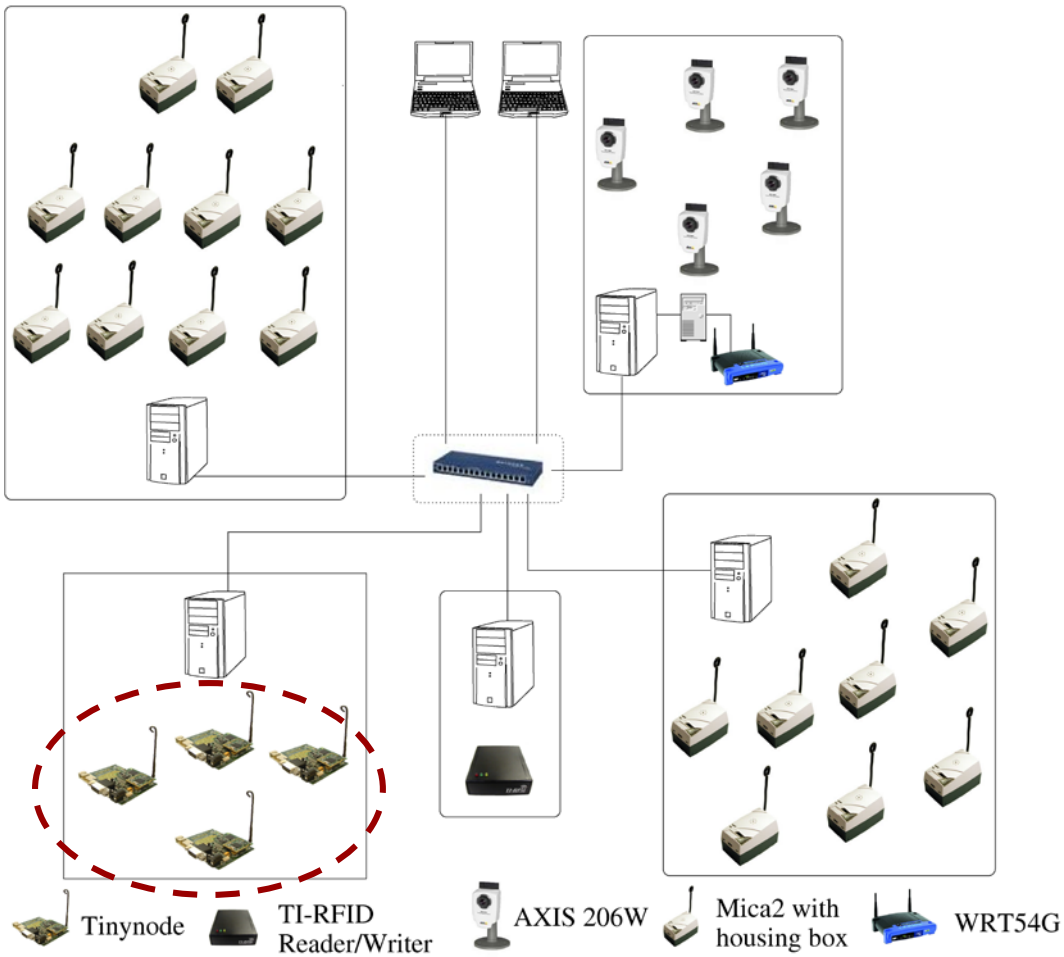
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18
- **SN-1:** 10 Mica2 motes with light and temperature sensors, packet size 15 Bytes, TinyOS

Experimental setup



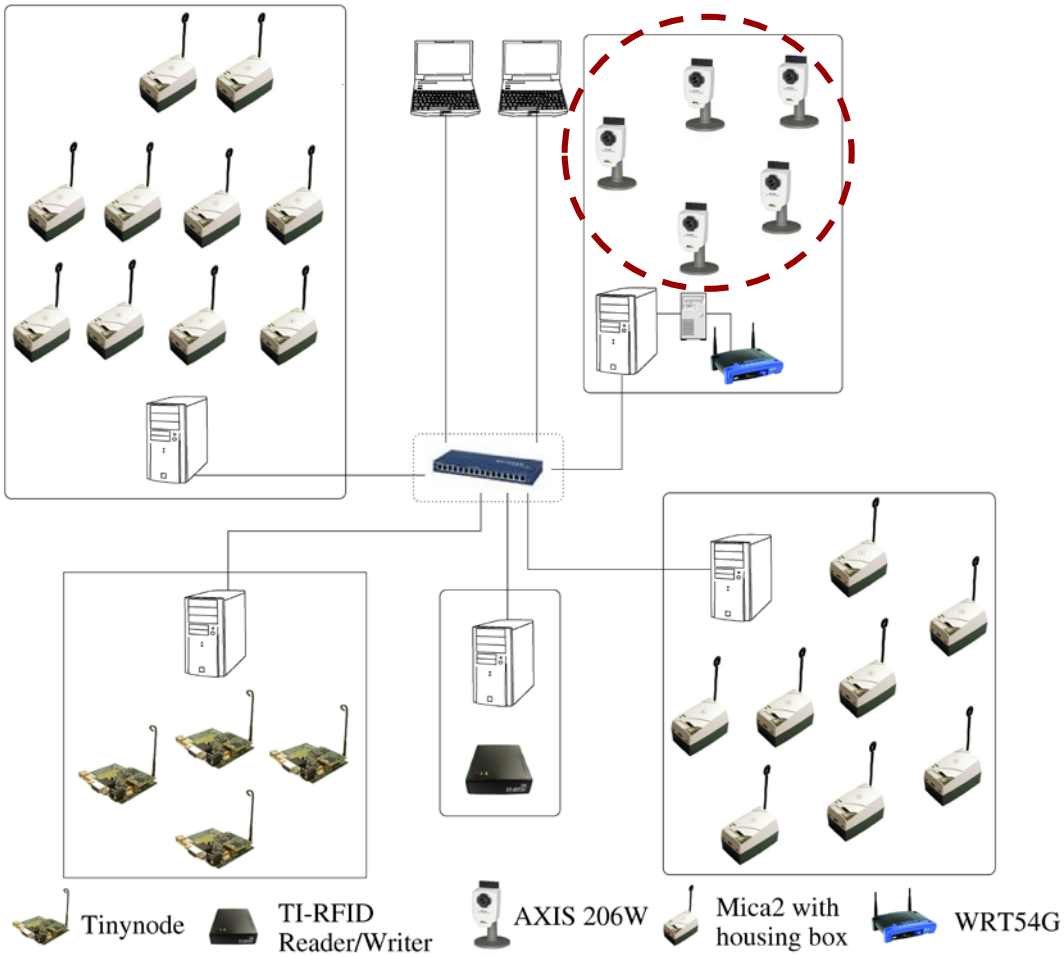
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18
- **SN-1:** 10 Mica2 motes with light and temperature sensors, packet size 15 Bytes, TinyOS
- **SN-2:** 8 Mica2 motes with light, temperature, acceleration, and sound sensors, packet size 100 Bytes, TinyOS

Experimental setup



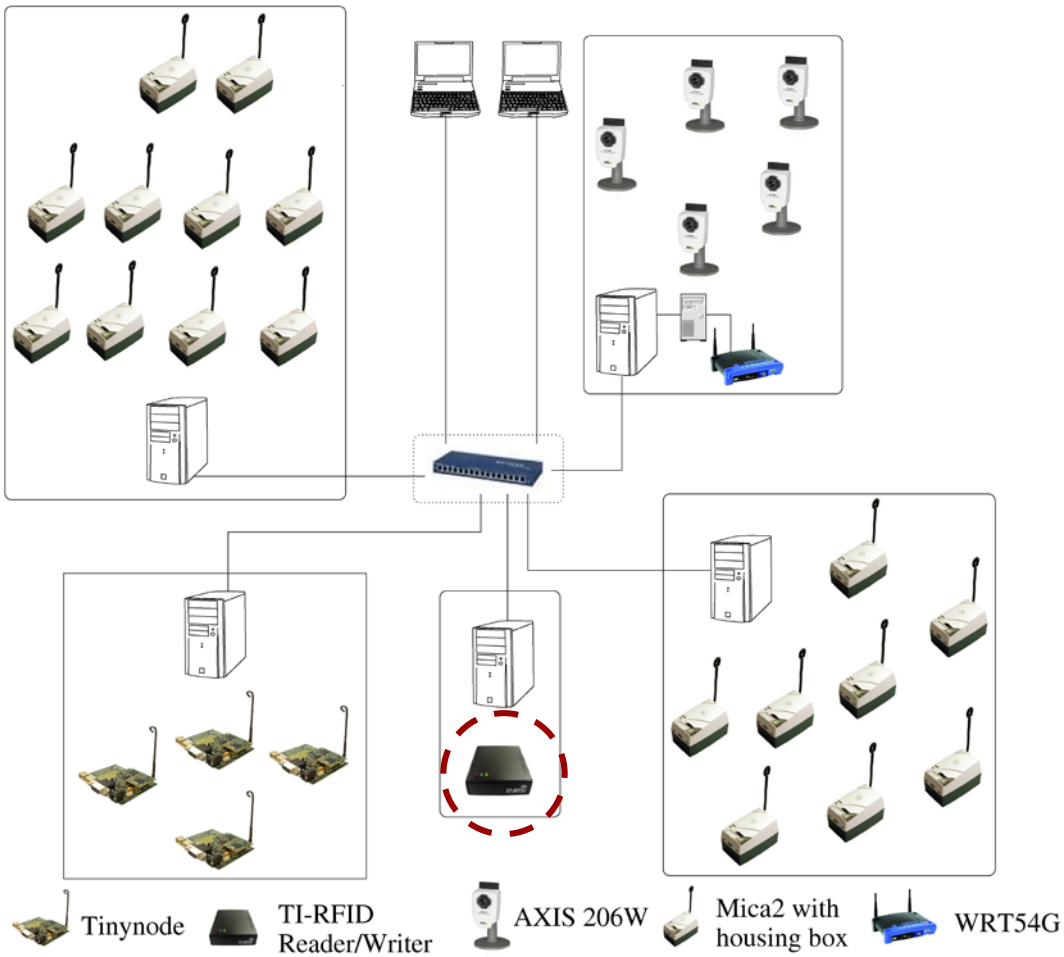
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18
- **SN-1:** 10 Mica2 motes with light and temperature sensors, packet size 15 Bytes, TinyOS
- **SN-2:** 8 Mica2 motes with light, temperature, acceleration, and sound sensors, packet size 100 Bytes, TinyOS
- **SN-3:** 4 Shockfish Tiny-Nodes with a light and two temperature sensors packet size 29 Bytes, TinyOS

Experimental setup



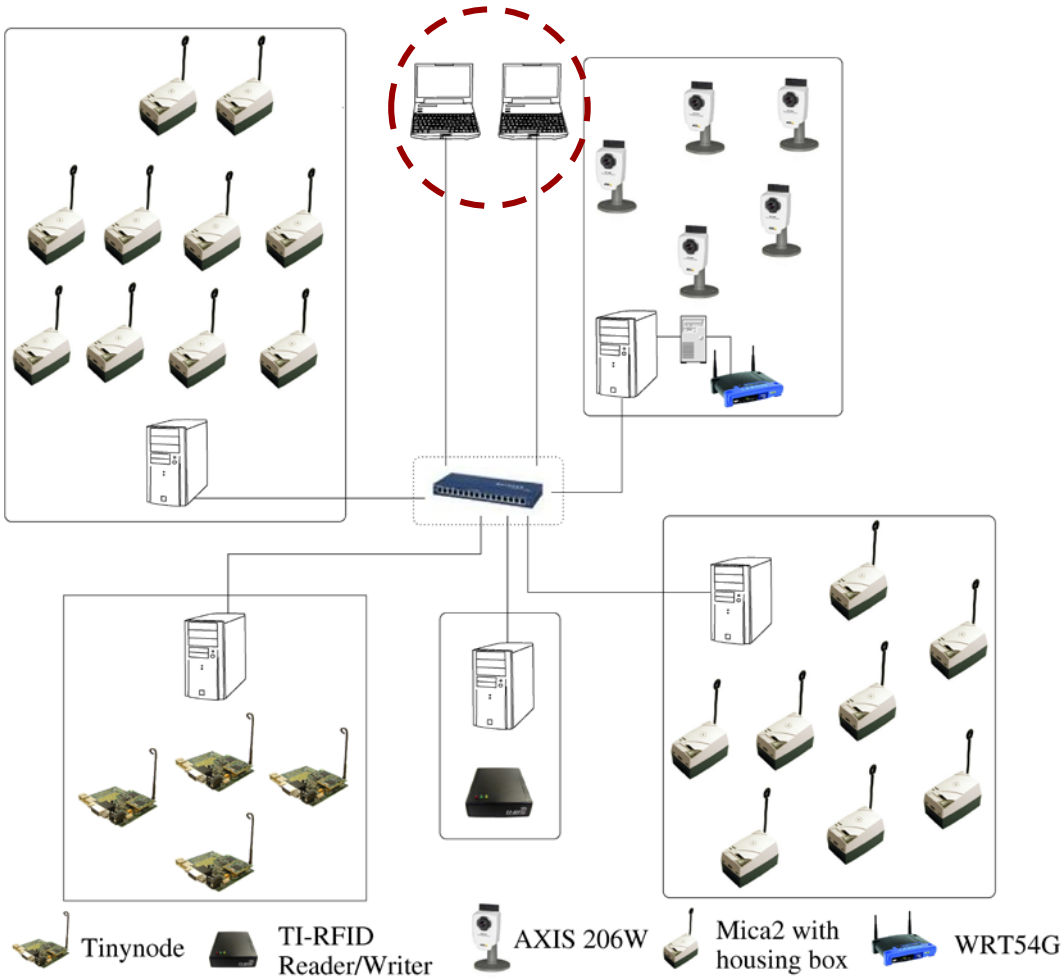
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18
- **SN-1:** 10 Mica2 motes with light and temperature sensors, packet size 15 Bytes, TinyOS
- **SN-2:** 8 Mica2 motes with light, temperature, acceleration, and sound sensors, packet size 100 Bytes, TinyOS
- **SN-3:** 4 Shockfish Tiny-Nodes with a light and two temperature sensors packet size 29 Bytes, TinyOS
- **SN-4:** 15 wireless 8002.11b cameras (AXIS 206W), 640x480 JPEG, 5 with 16kB average image size, 5 with 32kB, 5 with 75kB

Experimental setup



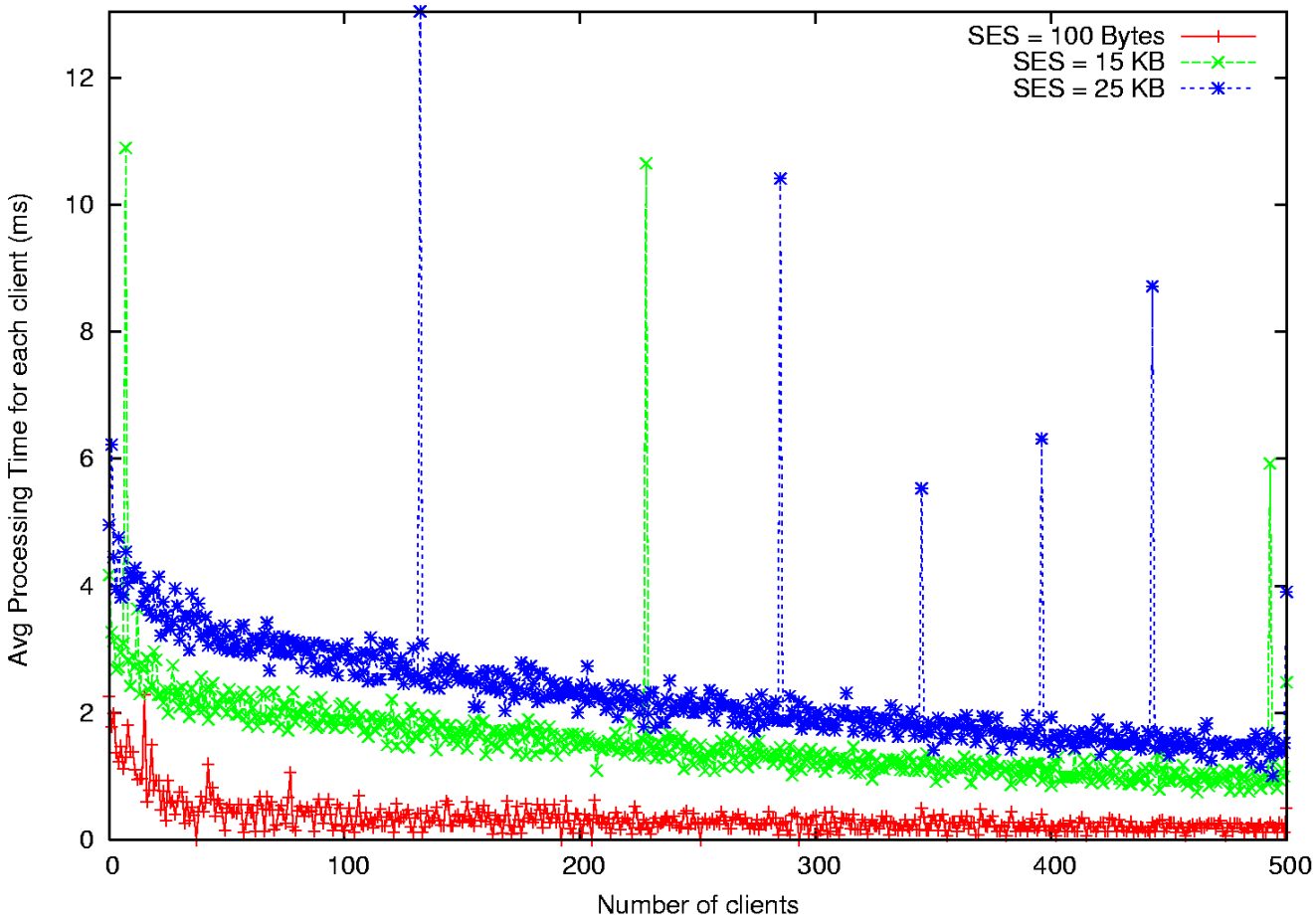
- **5 desktop PCs**
  - Pentium 4, 3.2GHz with 1MB cache, 1GB memory, 100Mbit Ethernet, Debian 3.1
  - Linux kernel 2.4.27, MySQL 5.18
- **SN-1:** 10 Mica2 motes with light and temperature sensors, packet size 15 Bytes, TinyOS
- **SN-2:** 8 Mica2 motes with light, temperature, acceleration, and sound sensors, packet size 100 Bytes, TinyOS
- **SN-3:** 4 Shockfish Tiny-Nodes with a light and two temperature sensors packet size 29 Bytes, TinyOS
- **SN-4:** 15 wireless 8002.11b cameras (AXIS 206W), 640x480 JPEG, 5 with 16kB average image size, 5 with 32kB, 5 with 75kB
- **SN-5:** TI Series 6000 S6700 multi-protocol RFID reader with three different kind of tags (up to 8KB of data)

Experimental setup

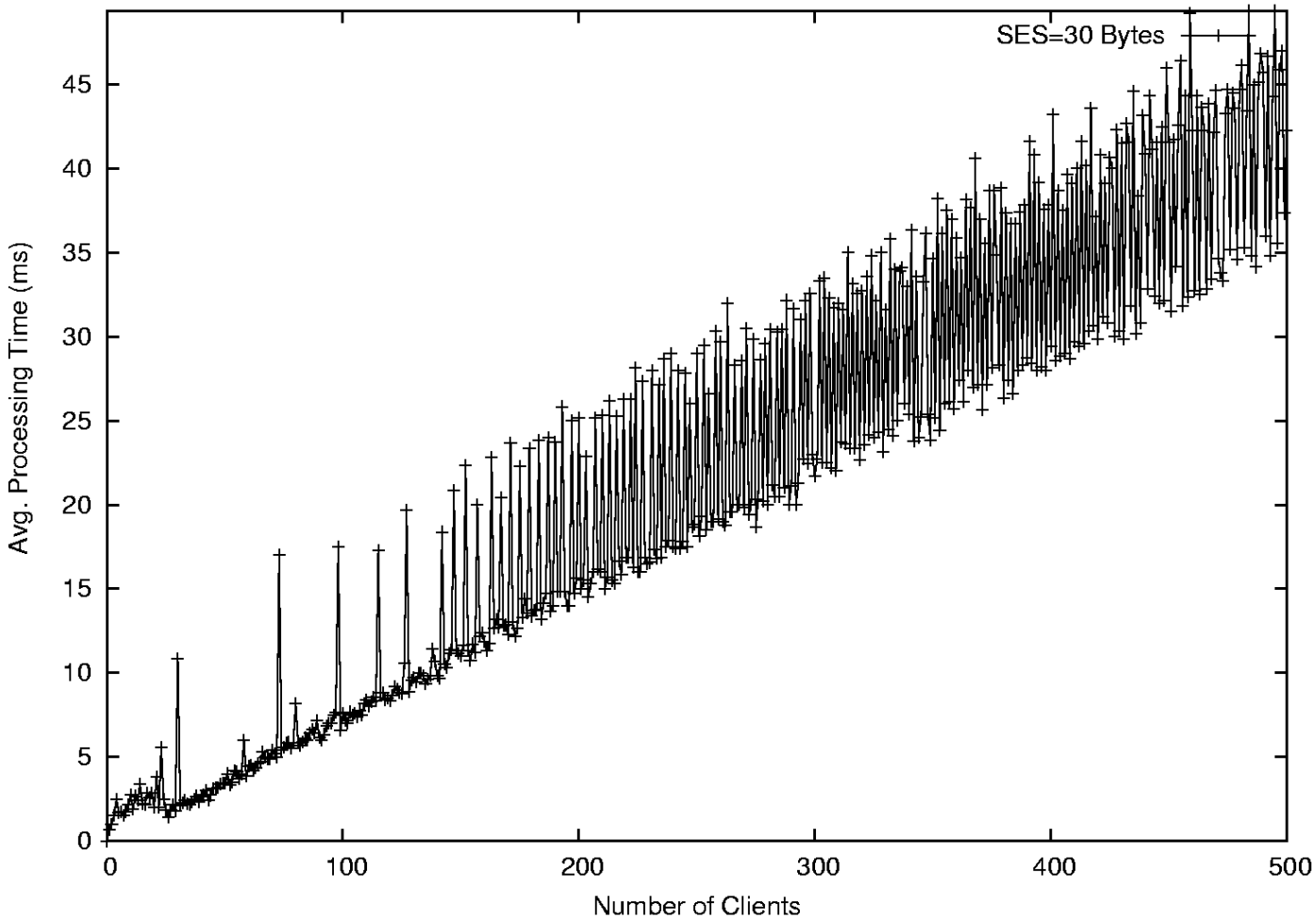


- 2 1.8 GHz Centrino laptops with 1GB memory as observers
- Each ran up to 250 lightweight GSN instances.
- Each instance produced random queries with varying table names, varying filtering condition complexity, and varying configuration parameters
- 3 filtering predicates in the WHERE clause on average, using random history sizes from 1 second up to 30 minutes and uniformly distributed random sampling rates (seconds) [0.01, 1]
- Motes produce random bursts (1-100 data items) with 25% probability

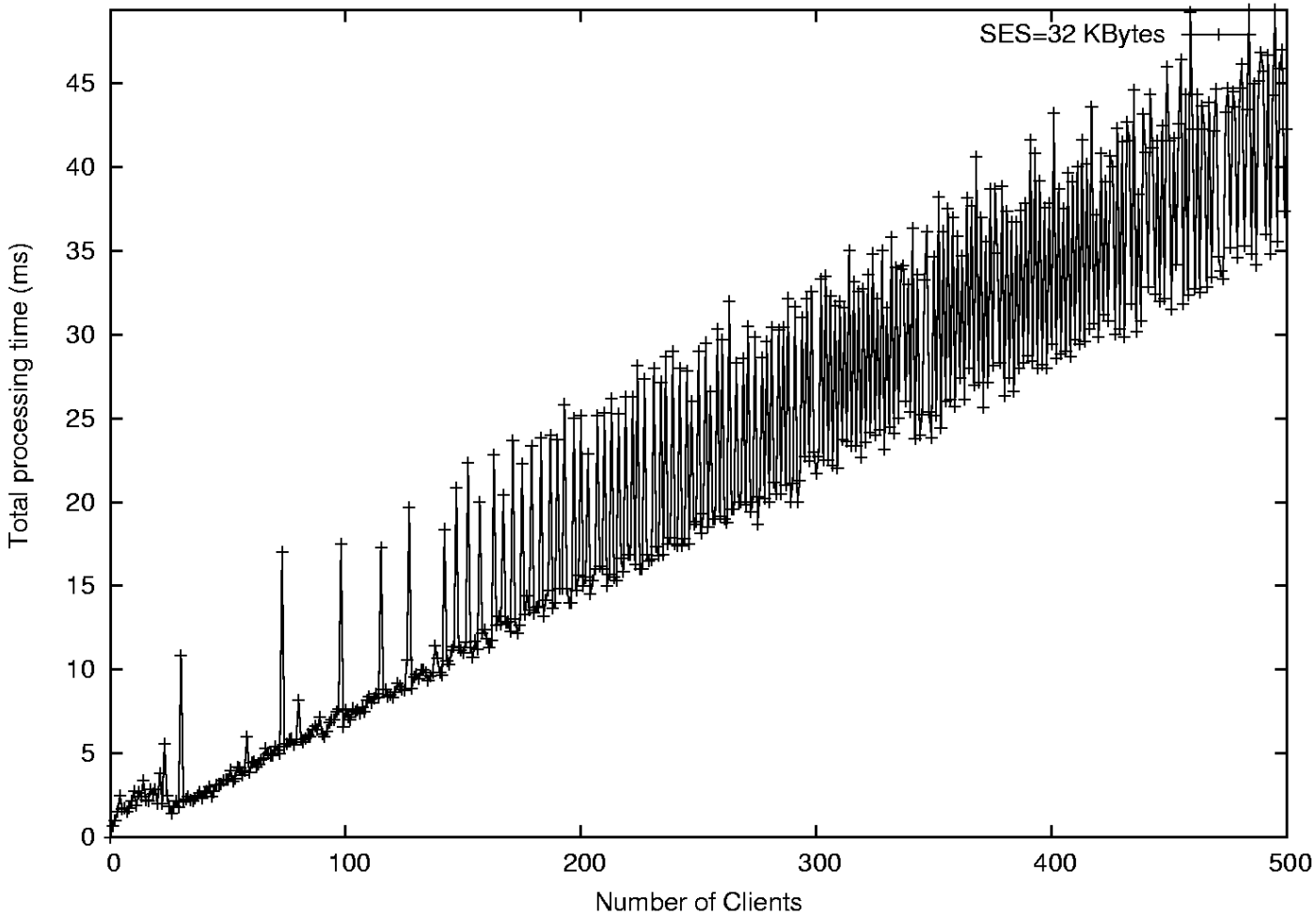
Processing time per client



Scalability in the number of clients (1)



Scalability in the number of clients (2)

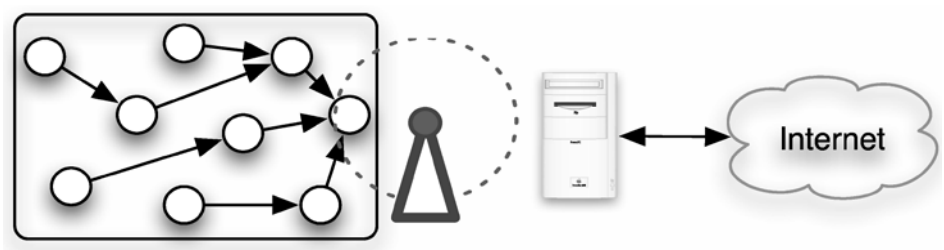


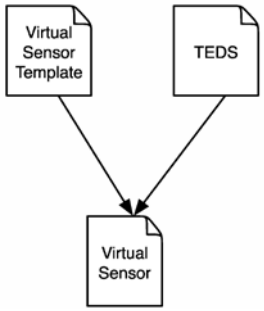
“Intelligent” infrastrcutures

or

The “Internet of Things”

Plug & play: Zero-programming deployment





- An IEEE 1451-compliant sensor provides a Transducer Electronic Data Sheet (TEDS) which is stored inside the sensor
- TEDS provides a simple semantic description of the sensor
  - the sensor's properties and measurement characteristic
- GSN uses the TEDS self-description feature for dynamic generation and deployment of virtual sensor descriptions
- Next step: **store queries not only data in TEDS or RFID tags**

⇒ New level of data processing in terms of flexibility

Where are we now?

- Simple declarative deployment based on XML
- Flexible query processing based on SQL
- Dynamic (re-)configuration
- Plug & play and zero-programming deployment
- Support for all major platforms
- Easy to extend
- Powerful abstractions and uniform interface
- Scalable P2P architecture
- Web service access and visualization

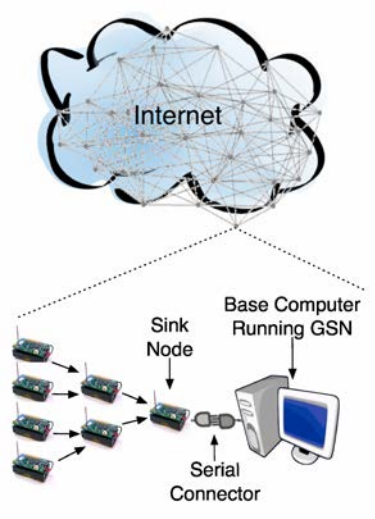
http://globalsn.sourceforge.net/

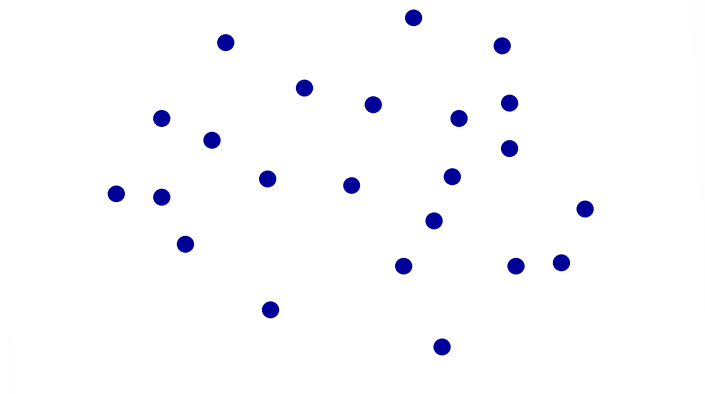
What is next?

- Handling and processing of very large amounts of temporary data
- Semantic descriptions of sensor networks and sensor data
- Semantic discovery
- Sensor data integration
- Large-scale deployments
  - e-Health
  - Smart homes
  - Tracking

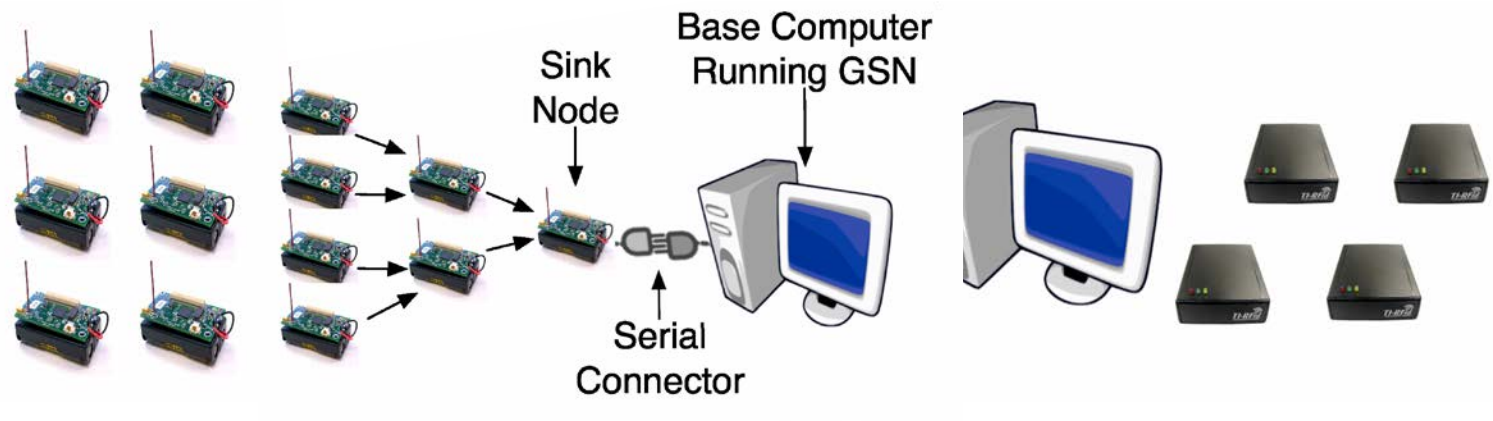
⇒ **Semantic** sensor networks

The “Sensor Internet” vision





- Sensor network + base computer = **sensor node**
- Many sensor nodes produce a lot of data on the Internet
- Questions:
  - Deployment
  - Description
  - Discovery
  - Integration
  - Distributed processing



Sensor Web Enablement WG

(SWE-WG)

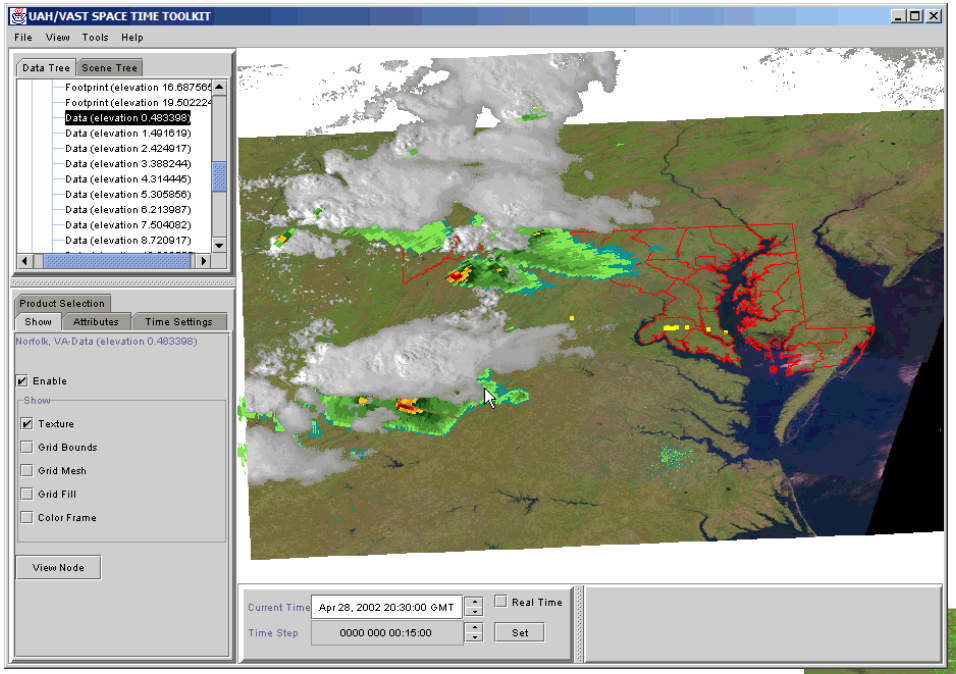
Open Geospatial Consortium


Sensors Are Everywhere

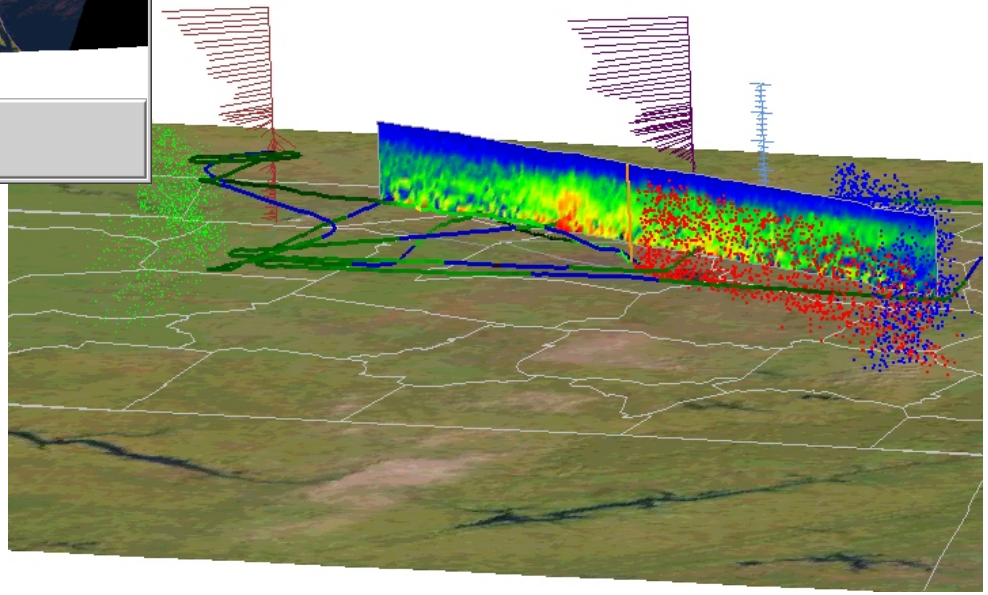
Sensor Web Desires

- Quickly **discover sensors** (secure or public) that can meet my needs – location, observables, quality, ability to task
- **Obtain sensor information** in a standard encoding that is understandable by me and my software
- Readily **access sensor observations** in a common manner, and in a form specific to my needs
- **Task sensors**, when possible, to meet my specific needs
- Subscribe to and **receive alerts** when a sensor measures a particular phenomenon

Sensor Web Desires



We desire the ability to discover and integrate observations from any sensor that meets our needs



Sensor Web Vision

Sensor Web Vision -1-

- Sensors will be web accessible
- Sensors and sensor data will be discoverable
- Sensors will be self-describing to humans and software (using a standard encoding)
- Most sensor observations will be easily accessible in real time over the web

Sensor Web Vision -2-

- Standardized web services will exist for accessing sensor information and sensor observations
- Sensor systems will be capable of real-time mining of observations to find phenomena of immediate interest
- Sensors will be capable of issuing alerts based on observations, as well as be able to respond to alerts issued by other sensors

Sensor Web Vision -3-

- Sensors and sensor nets will be able to act on their own (i.e. be autonomous)
- Software will immediately be capable of geolocating and processing observations from a newly-discovered sensor

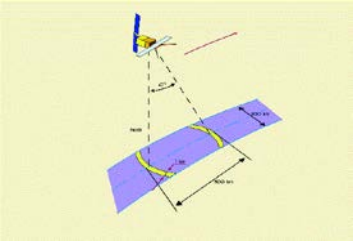
OpenGIS Sensor Web Enablement

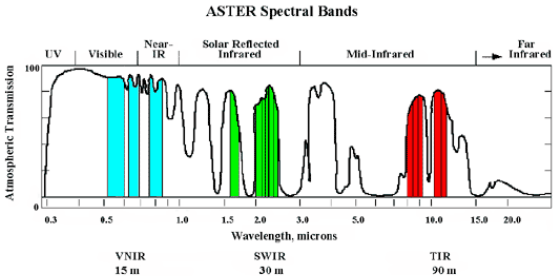
OGC Sensor Web Enablement -1-

- "High-level" OGC Services supporting sensor data
  - **Web Map Service** – request map (i.e. image)
  - **Web Coverage Service** – request binary gridded or aggregated data
  - **Web Feature Service** – request feature data

OGC Sensor Web Enablement -2-

- Sensor Web Enablement Framework - Schema
  - **SensorML –** models and schema for describing sensor characteristics (geolocation, response)





  - **Observation & Measurement** – models and schema for encoding sensor observations

OGC Sensor Web Enablement -3-

- Sensor Web Enablement Framework – Services
  - **Sensor Observation Service** – access sensor information (SensorML) and sensor observations (O&M)
  - **Sensor Planning Service** – task sensors or sensor systems
  - **Web Alert Service** – asynchronous notification of sensor events (tasks, observation of phenomena)
  - **Sensor Registries** – discovery of sensors and sensor data

OGC Sensor Web Enablement -4-

- Sensor Web Enablement Framework – Probable Additions
  - **TransducerML** – XML protocol for streaming data clusters from transducers (sensors, transmitters, actuators)
    - currently combining with SensorML
  - **Common Alert Protocol (CAP)** – developed by international emergency management community for XML publishing of events

OGC Sensor Web Enablement Status

- Framework is well-thought out and compatible with Enterprise Engineering concepts
- All components are real, but some components more mature than others
- SensorML is targeted for OGC approval in Fall 2004
- A list serve / forum is established for SensorML
- Minor level of funding would allow Interoperability Testing and initial implementation by vendors and data suppliers

Sensor Model Language
(SensorML)

Overview of Concepts and Applications

SensorML History -1-

- NASA Interuse Experiment (UAH, JPL: ‘93–’96)
  - **Use of SPICE planetary navigation system for Earth Observation**
  - **Initial Sensor description file developed by Bill Taber of JPL**

- Space Time Toolkit (UAH: ’95 - present)
  - **Originally utilized SPICE for geolocation**
  - **Later developed light-weight JAVA-based geolocation system with hardwired sensor classes**
  - **Being used to test prototype SensorML designs through the application of sensor model classes**

SensorML History -2-

April 1998 - Recommendation to CEOS: Interoperability of Multiple Space Agency Sensor Data from the Global Mapping Task Team – Bernried, Germany

Definition of Problem: **There is an increasing realization by Earth observation scientists that data from space-borne sensors are not adequately nor easily georeferenced to meet their requirements. The consequence of this is that it is currently extremely difficult or impossible to combine data from different space-borne sensors or ground-based data. The first impediment is the lack of adequate, publicly available data on the spatial-temporal extents of data from space-borne sensors.**

SensorML History -2-

Recommendation: **We, the CEOS Global Mapping Task Team, recommend that the space agencies seriously consider the production, storage, public access, and interoperability of adequate data for describing the dynamics and geometry of the sensor system. These data might include satellite position (ephemeris), satellite rotations (attitude), sensor model (dynamics, geometry, and calibration), relevant planet models, and spacecraft clock model. This data should be made available in real-time.**

There should also be an effort to provide publicly available software to ingest the above data using a common API. Any recommendations for the most appropriate propagation model should be adequate documented or algorithms provided.

SensorML History -3-

- CEOS Global Mapping Task Team – Boulder (Botts – Sept. 1998)
  - **Presents beginnings of a Sensor Description Format**
  - **Task Team recommends consideration of XML**

- CEOS Global Mapping Task Team – Charlottesville (Botts – Sept. 1999)
  - **Presents the initial XML version of SensorML**

- NASA Advanced Information Systems Technology (AIST) Program – March 2000
  - **March 2000 - Botts proposal accepted to continue development and implementation of SensorML**
  - **Dec 2000 – funding received**

SensorML History -4-

- ISO TC 211, Project 19130 – Sensor and Data Model
  - **Dec 2000 - Liping Di (CEOS GMTT member) proposes and gets accepted Sensor Model standards study**
  - **Botts asked to serve as Team Member (he hasn't been much help so far)**

- OpenGIS Military Pilot Project (MPP1) – March 2001
  - **Botts accepted to participate with emphasis on Sensor Web enabling**

SensorML History -4-

- OpenGIS Open Web Services (OWS1) – September 2001
  - **Botts accepted to participate in defining OGC Sensor Web standards and protocols**
  - **SensorML both serving as a prototype model and undergoing significant extensions and modifications under activity … particularly for more robust support of in-situ sensors**
  - **EPA sponsorship**

- OpenGIS OWS 1.2 – May 2002
  - **Continuation of SensorML development and incorporation into Sensor Web Enablement with emphasis on dynamic, remote sensing**
  - **NIMA sponsorship**

OpenGIS Sensor Web Enablement Components

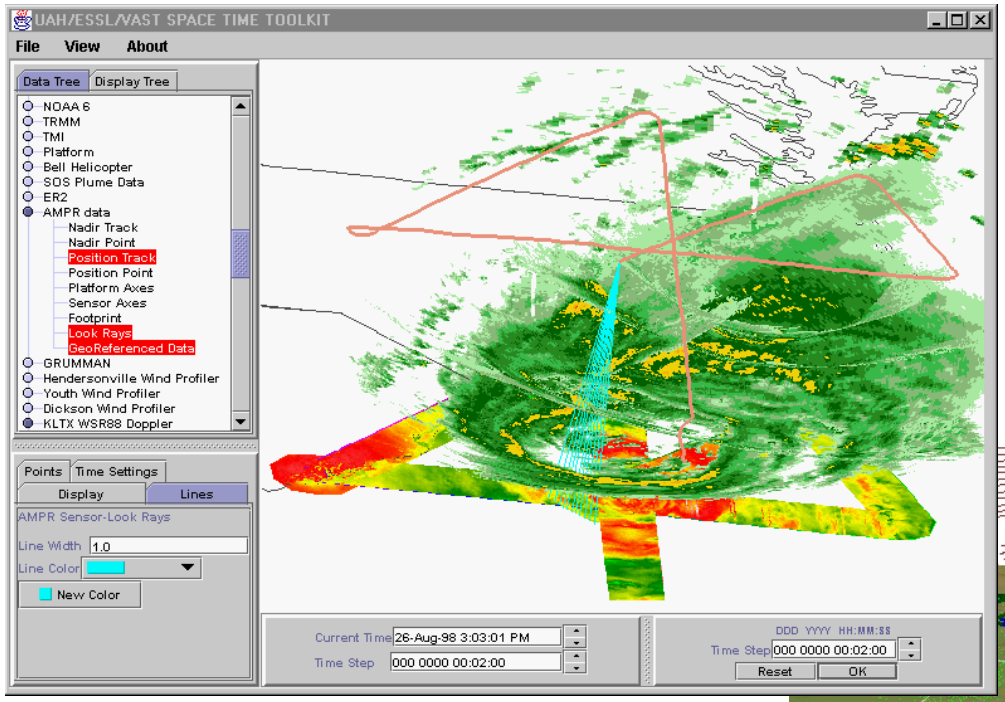
- Observable – **a phenomenon that can be observed and measured (referenced in an ObservablesDictionary)**
- Sensor – **a device that observes and/or measures a phenomenon (SensorML)**
- Observed value (Observation/Measurement) – **the value returned by or derived from a sensor observation (e.g. quantity, count, boolean, category, ordered category, position)**
- Sensor Collection Service – **a service that provides observed values**
- Registries – **provide discovery mechanism for sensors and observed values**

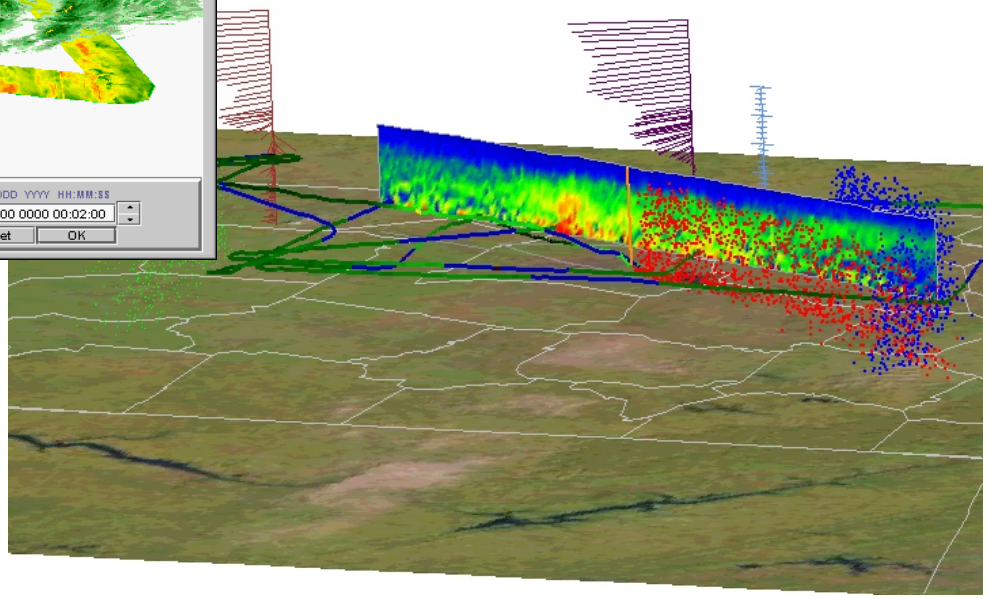
Scope of SensorML Support

- Designed to support a wide range of sensors
  - Including both dynamic and stationary platforms
  - Including both in-situ and remote sensors

- Examples:
  - Stationary, in-situ **– chemical "sniffer", thermometer, gravity meter**
  - Stationary, remote **– stream velocity profiler, atmospheric profiler, Doppler radar**
  - Dynamic, in-situ **– aircraft mounted ozone "sniffer", GPS unit, dropsonde**
  - Dynamic, remote **– satellite radiometer, airborne camera, soldier-mounted video**

Use of SensorML for a variety of sensor types





Information Provided by SensorML

- Observation characteristics
  - **Physical properties measured (e.g. radiometry, temperature, concentration, etc.)**
  - **Quality characteristics (e.g. accuracy, precision)**
  - **Response characteristics (e.g. spectral curve, temporal response, etc.)**

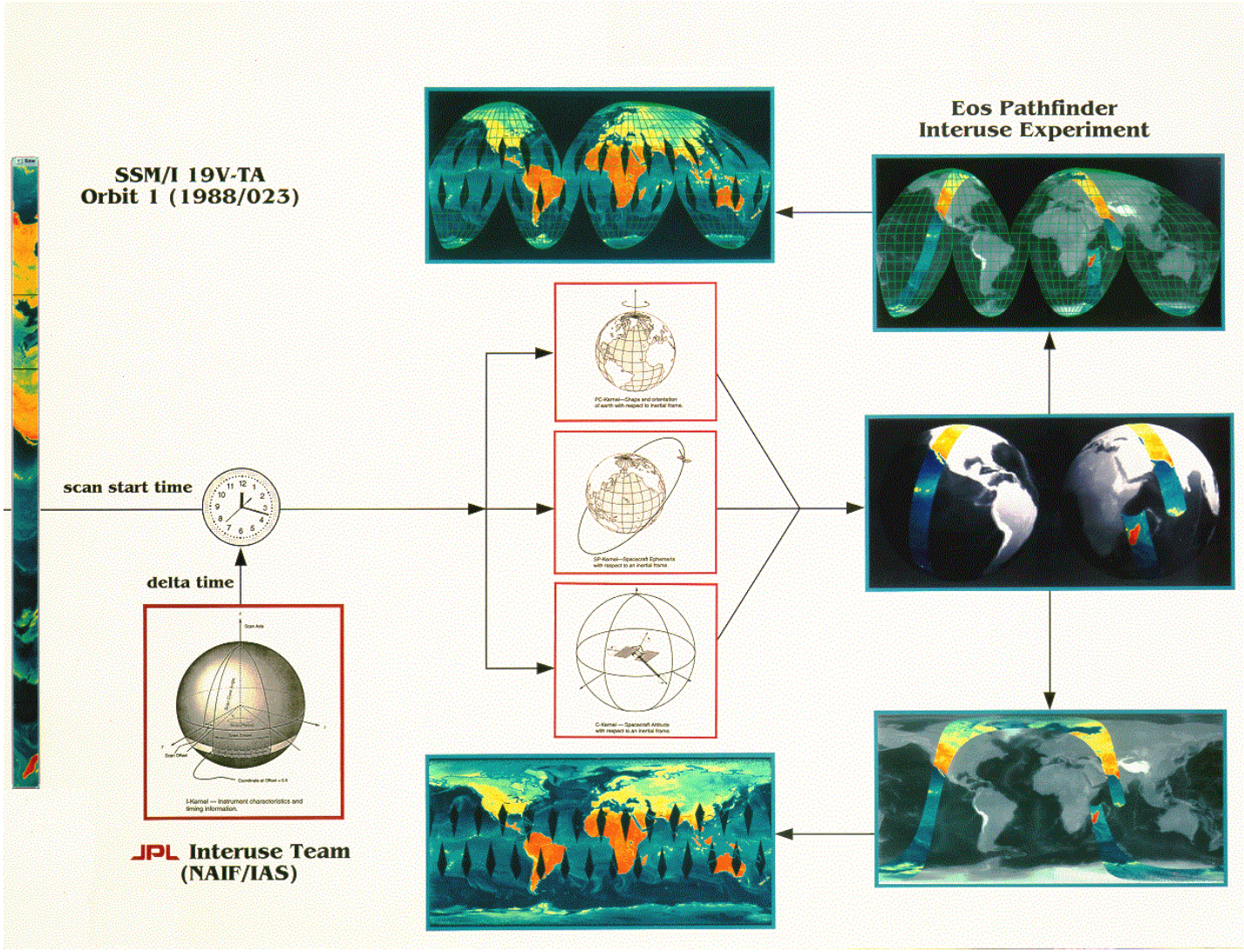
- Geometry Characteristics
  - **Size, shape, spatial weight function (e.g. point spread function) of individual samples**
  - **Geometric and temporal characteristics of sample collections (e.g. scans or arrays)**

- Description and Documentation
  - **Overall information about the sensor**
  - **History and reference information supporting the SensorML document**

SensorML Concepts

- SensorML is an XML schema for defining the geometric, dynamic, and observational characteristics of a sensor
- The purpose of the sensor description:
  - (1) provide general sensor information in support of data discovery
  - (2) support the processing and analysis of the sensor measurements
  - (3) support the geolocation of the measured data.
  - (4) provide performance characteristics (e.g. accuracy, threshold, etc.)
  - (5) archive fundamental properties and assumptions regarding sensor
- SensorML provides functional model for sensor, not detail description of hardware
- SensorML separates the sensor from its associated platform(s) and target(s)

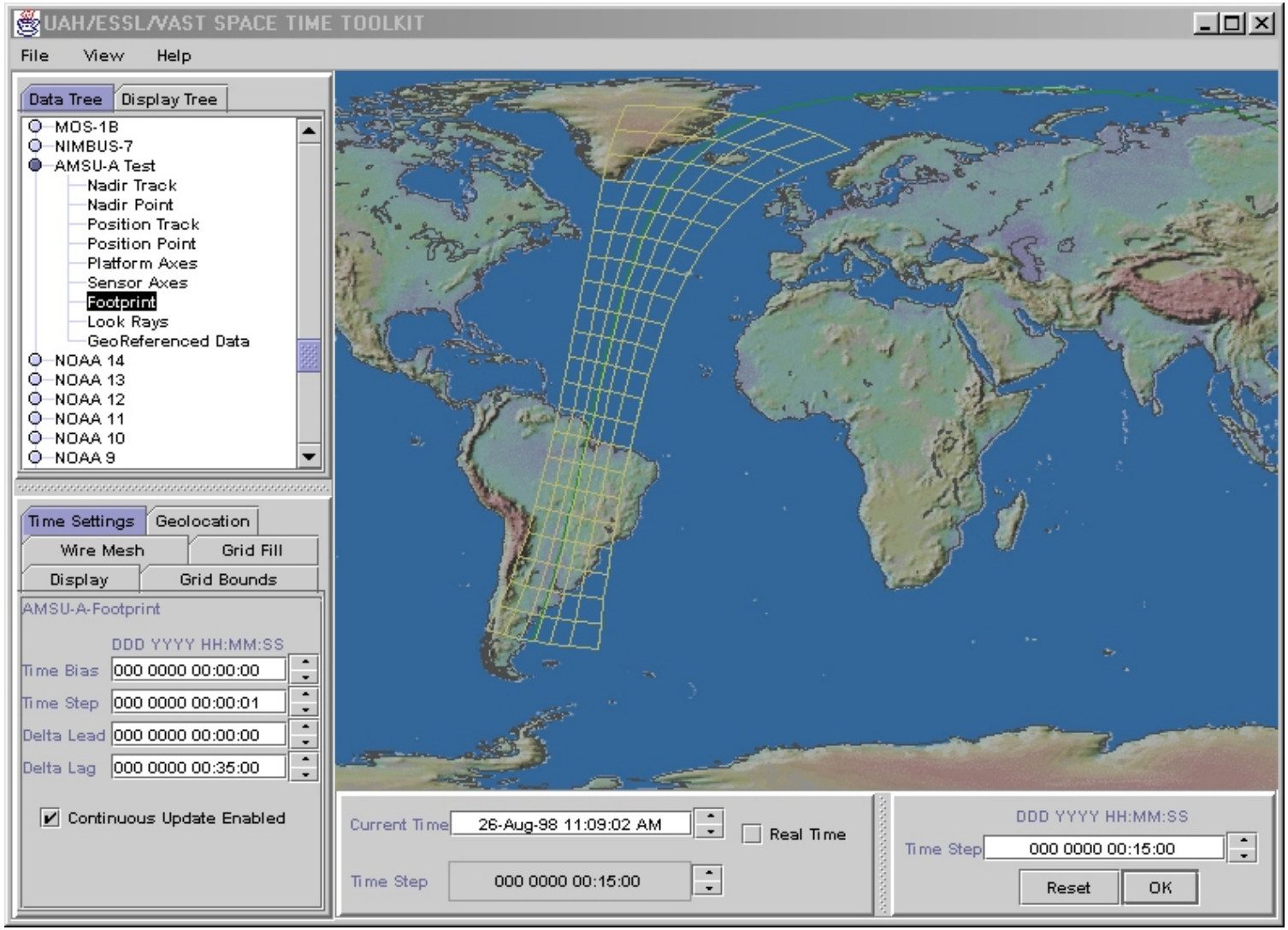
Components Needed for Geolocation of Sensor Data



Applications of SensorML and SensorWeb Technology

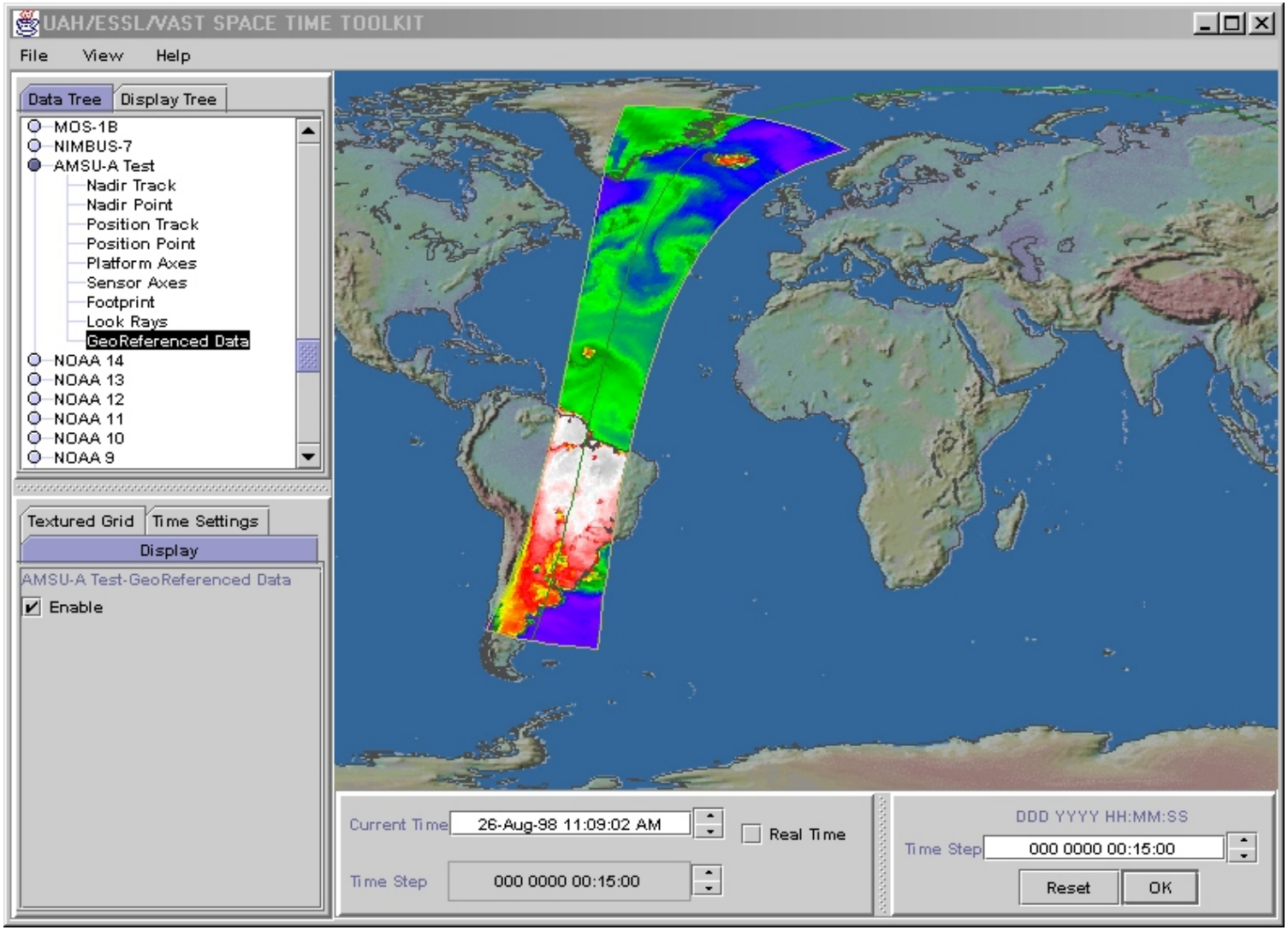
- **Coincident search for relevant data (i.e. data discovery)**
- **On-demand processing of data products**
- **Dynamic on-demand fusion of disparate sensor data (interuse)**
- **Pre-mission planning**
- **Real-time guidance and pointing (e.g. UAVs)**
- **On-board applications**
  - **Autonomous operation / target recognition**
  - **SensorWeb communication of location and targets**
  - **Direct transmission of data and processing information to remote sites**

Determination of Sensor Footprint with Time



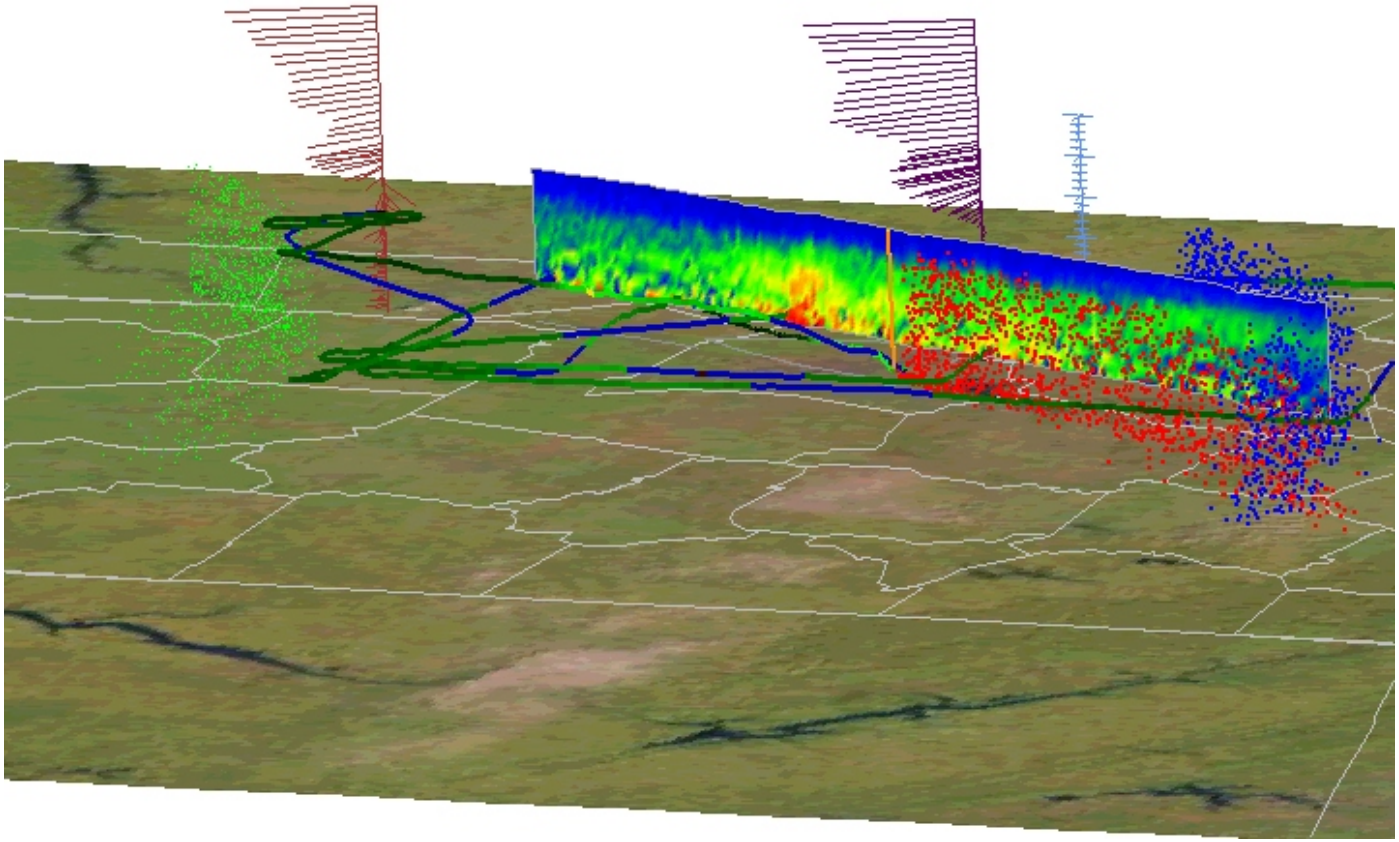
http://vast.uah.edu/SensorML/home.html

Intelligent Retrieval and Co-registration of Sensor Data



http://vast.uah.edu/SensorML/home.html

Visual Fusion of Disparate Sensor Data



http://vast.uah.edu/SensorML/home.html

SensorML Status

Completed Task and Current SensorML

Completed Tasks

COMPLETE:

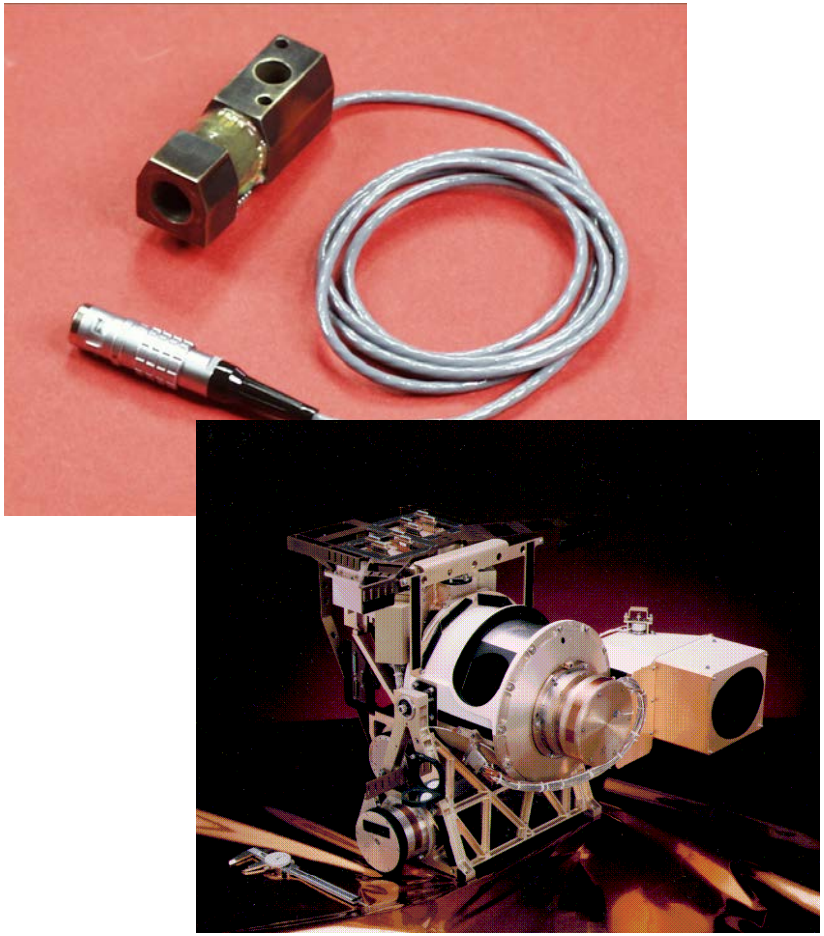
- Define the initial SensorML standard and evaluate robustness of design (03/01)
- Complete initial documentation
- Tested SensorML conceptual design for geolocation within hardwired sensor classes in Space Time Toolkit

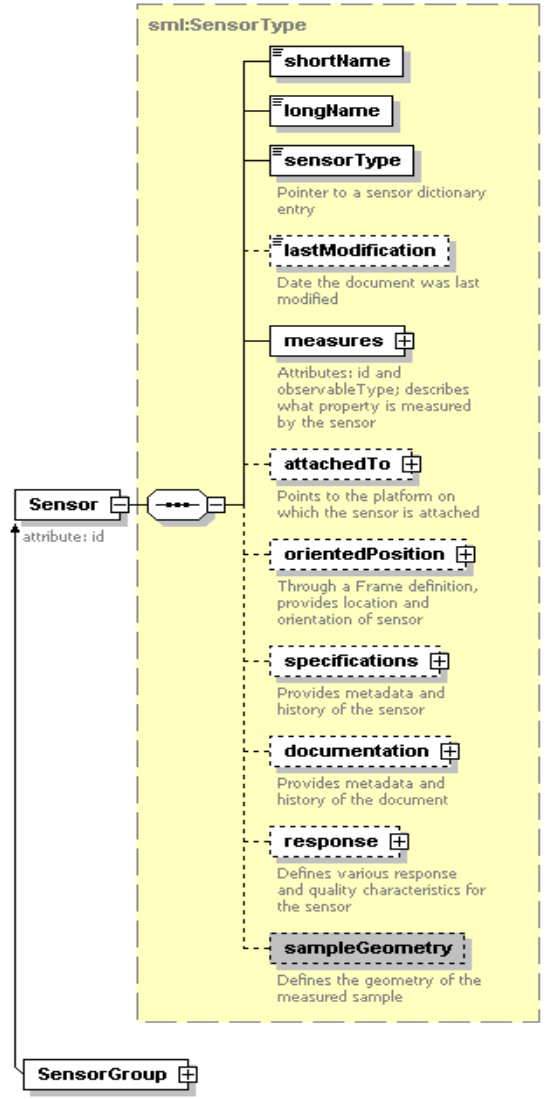
Completed Tasks

COMPLETE:

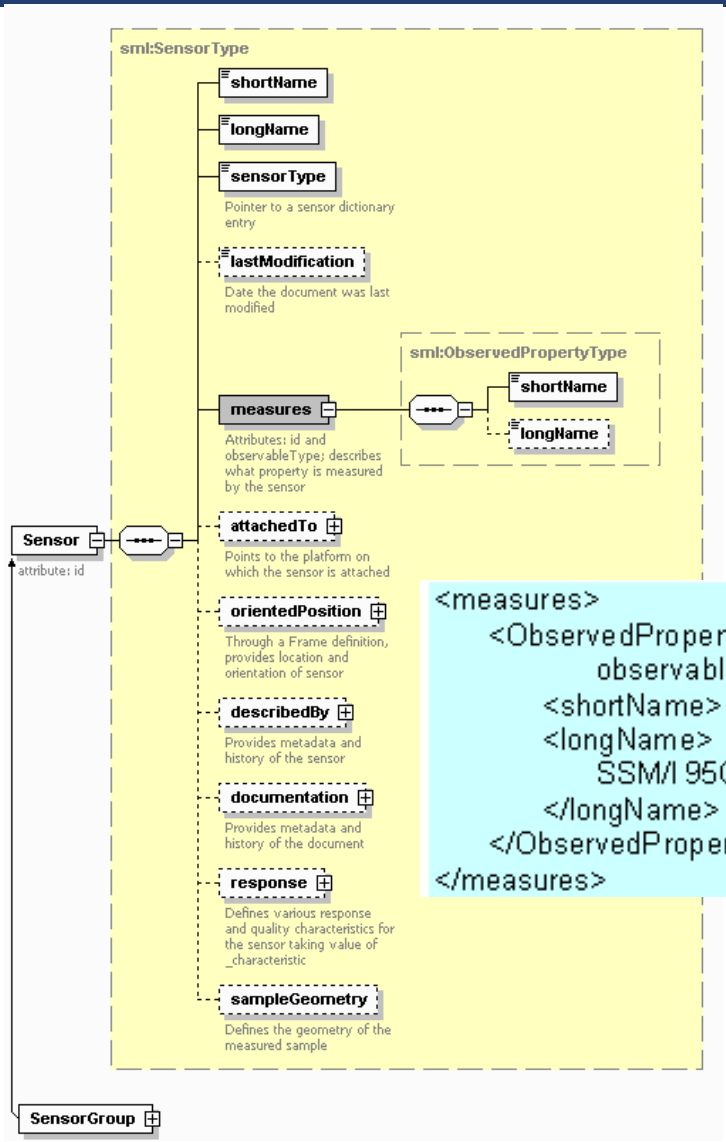
- Completed initial redesign of SensorML with several extensions and modifications
  - Redefined standard in XML schema (rather than DTD)
  - Provided more robust support for description of observation properties
  - More generalized support provided for in-situ sensors
  - Added capabilities for making SensorML instance a “living document” of sensor history and modifications – established Action schemas for both sensor and document history
  - Established Interoperability with OpenGIS standards, particularly Geographic Markup Language (GML), Sensor Collection Service (SCS), Observables, Sensor Registries
- Documented redesigned standard [OGC IPR 02-026]
- Current SensorML design implemented and tested under OpenGIS OWS1 activities

Basic Sensor Schema



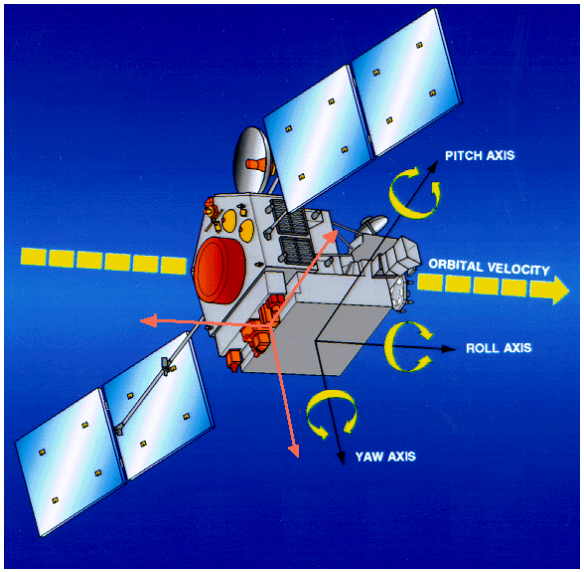


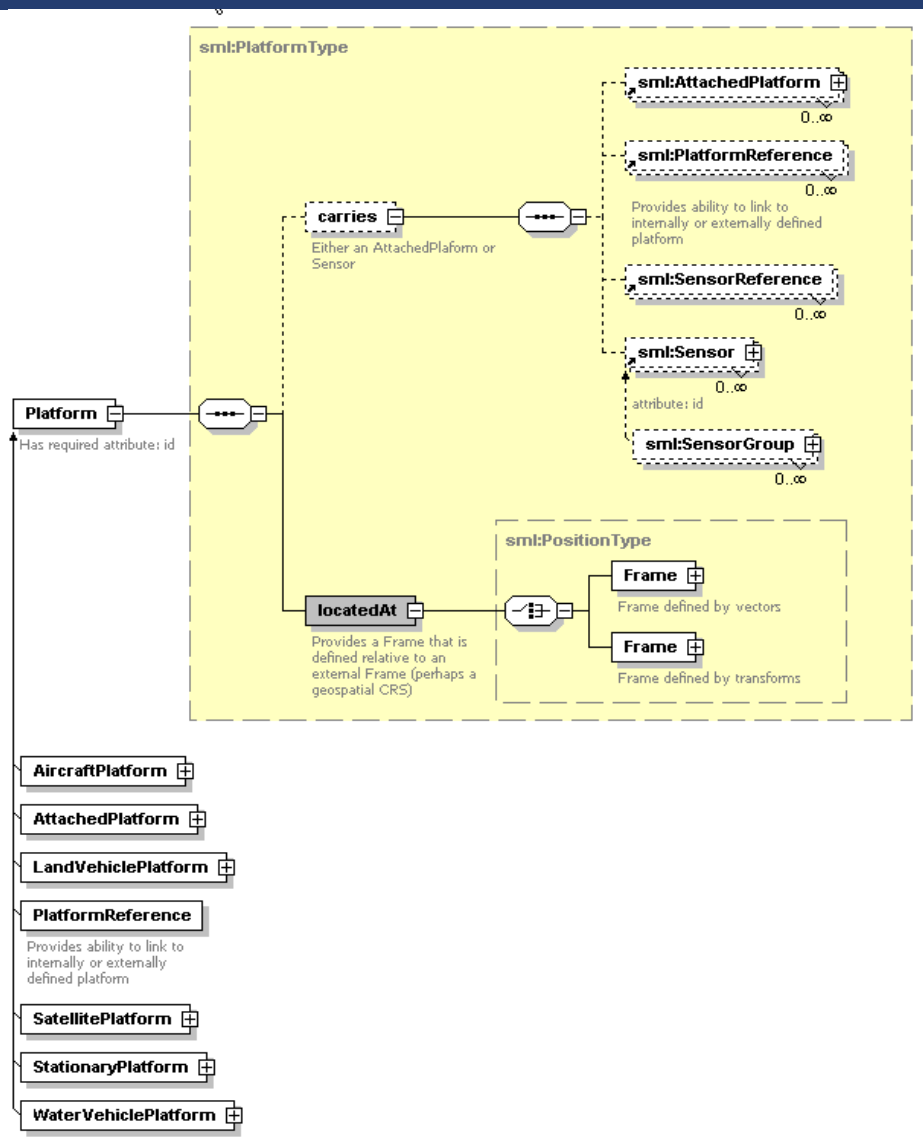
“measures” property



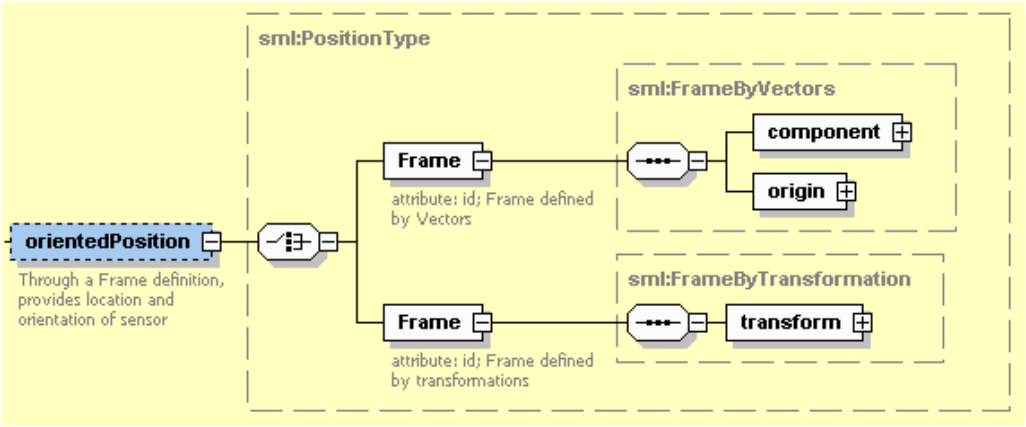

```
<measures>
    <ObservedProperty id="ssmi_f10_95V"
            observableType="http:www.opengis.net/observation/type#radiation">
        <shortName> 95 GHz V </shortName>
        <longName>
            SSM/I 95GHz vertical polarization channel on DMSP F10 satellite
        </longName>
    </ObservedProperty>
</measures>
```

Platform Schema



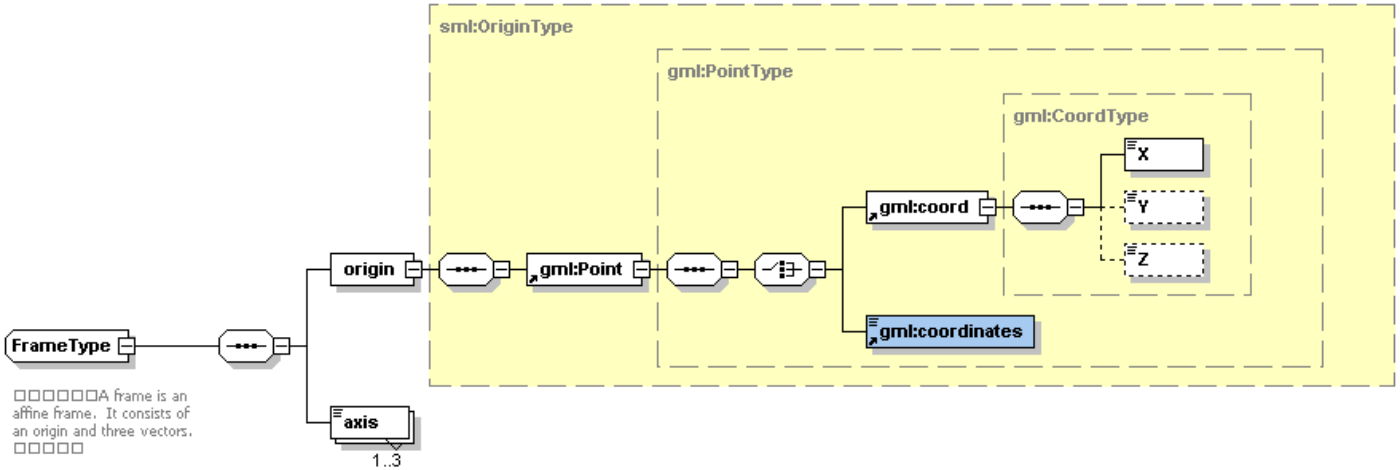


Frames and Oriented Position Schema

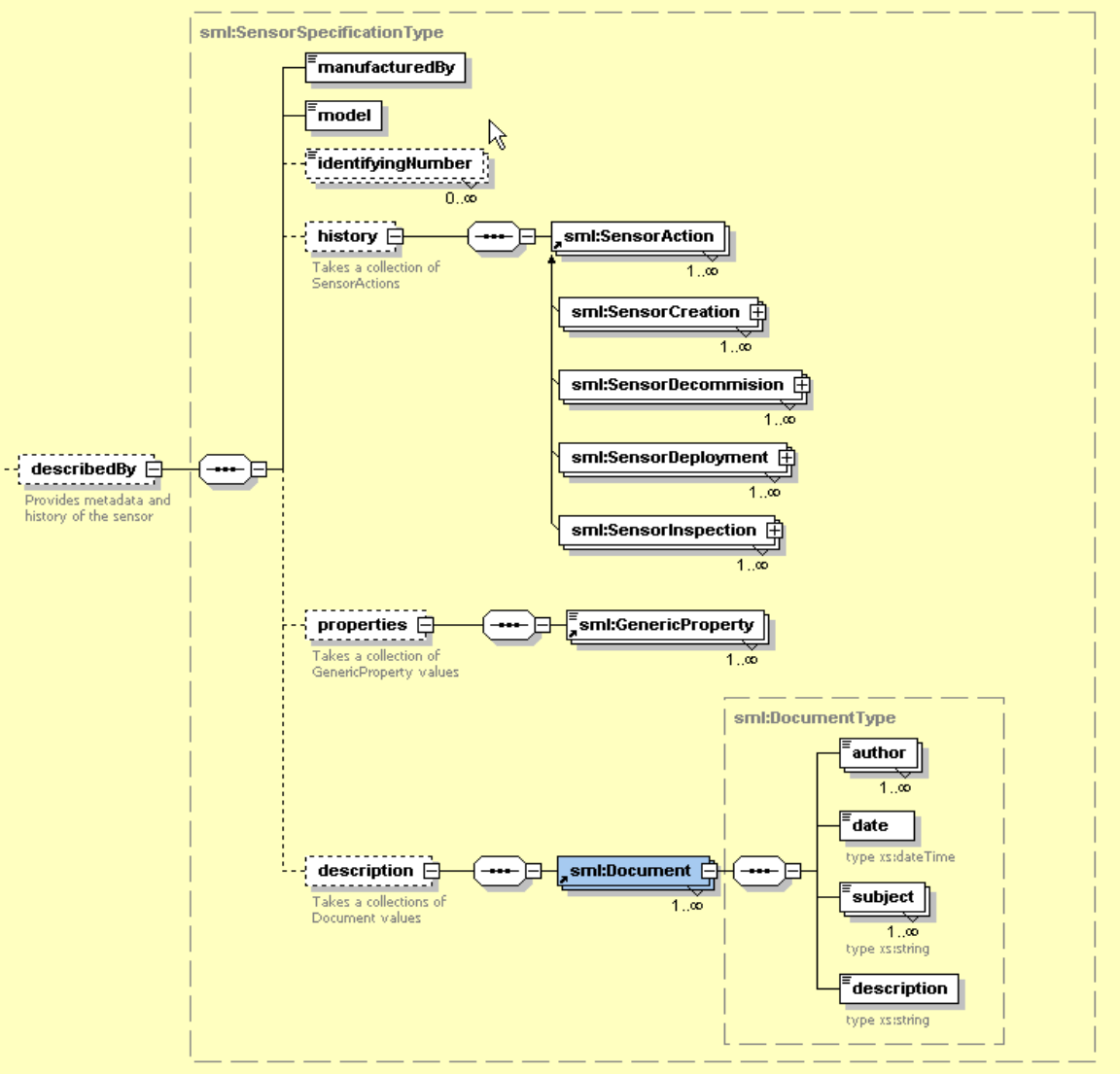


General Frame

GML Frame



Sensor Description

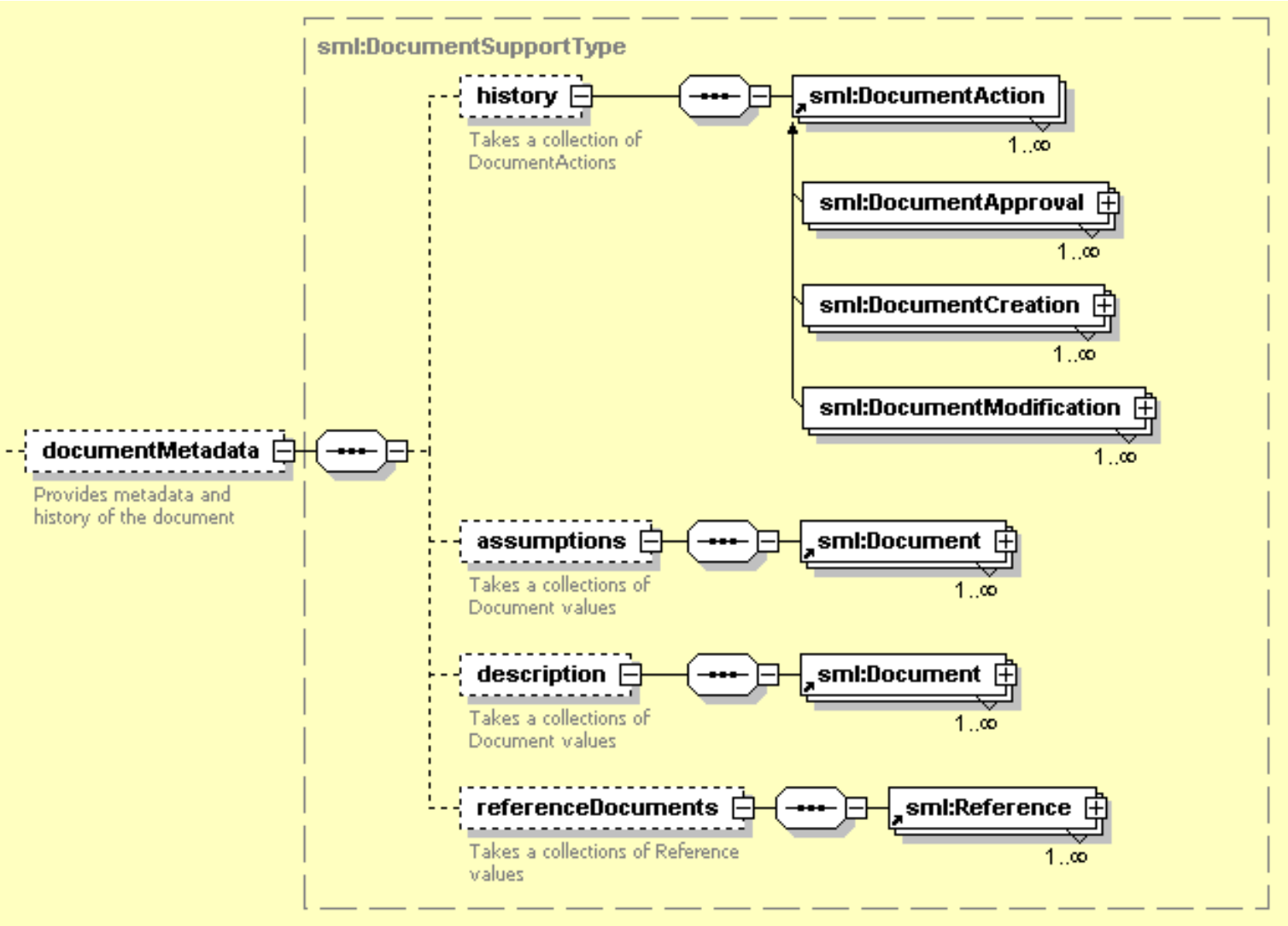


“describedBy” Property

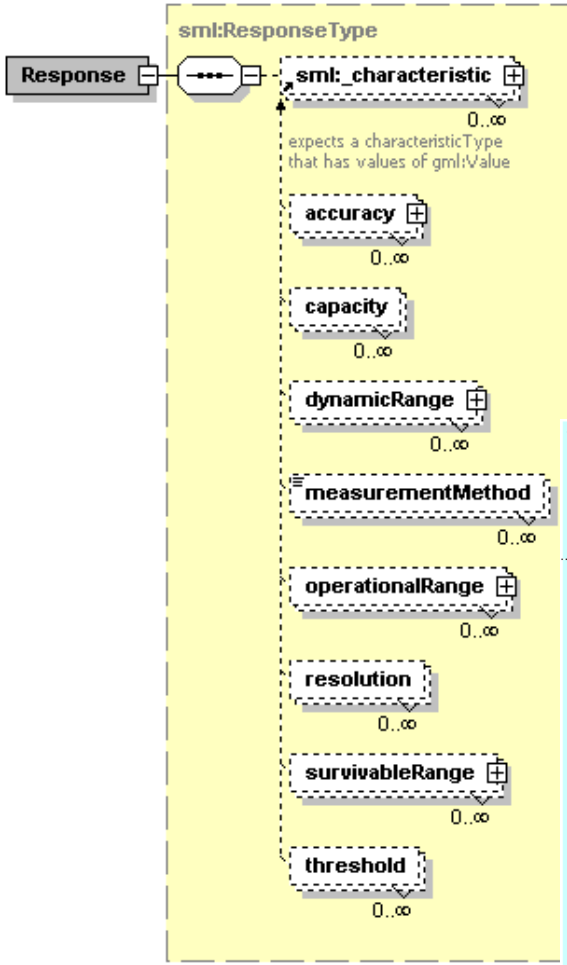
```
<genericProperty name="membrane thickness" dataType=" float"
    uom="uom:mil">1.0</genericProperty>
<genericProperty name="membrane type" dataType="string">Teflon</genericProperty>
<genericProperty name="probe solution type" dataType="chemical species"> Na2SO4
    </genericProperty>
```

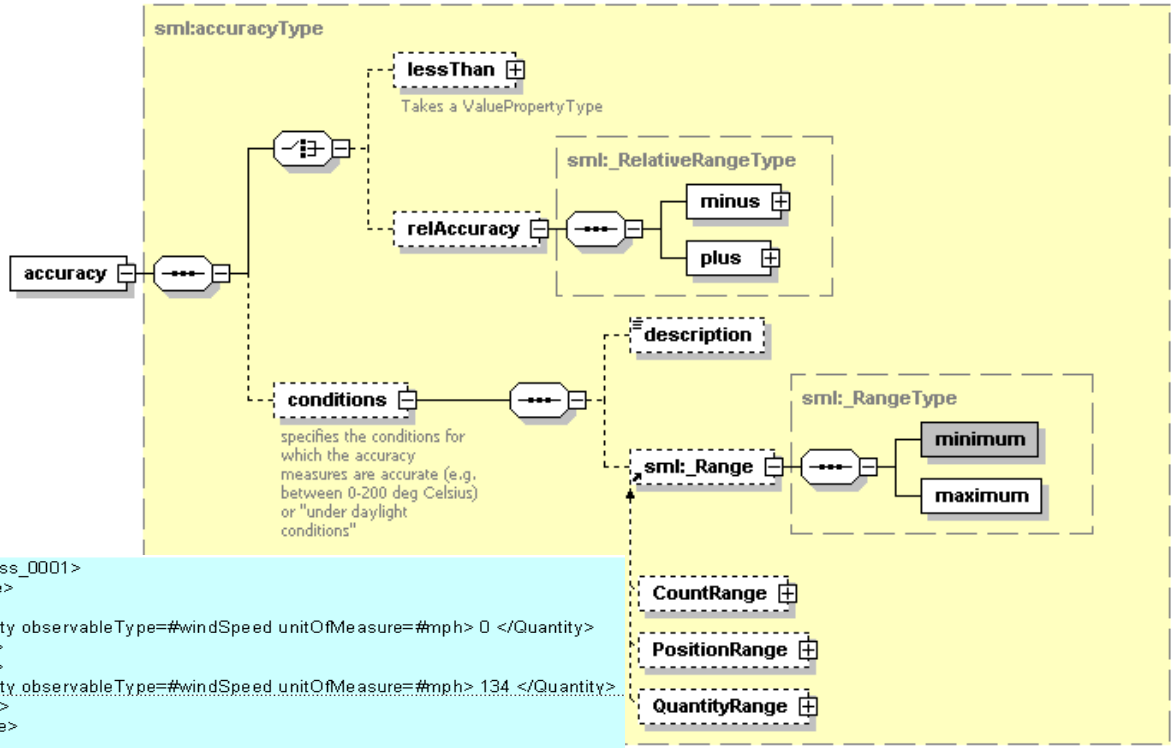
```
<describedBy>
    <SensorSpecification>
        <manufacturedBy> YSI </manufacturedBy>
        <model> 58 </model>
        <identifyingNumber type="serialNumber"> s454165 </identifyingNumber>
        <history>
            <SensorCreation>
                <byWhom>
                    <Person>
                        <fullName> Some Guy </fullName>
                    </Person>
                </byWhom>
                <when> 2001-12-04 </when>
                <supportingDocuments>
                    <Reference id="YSI-12">
                        <authors>
                            <Person>
                                <fullName> Some Author </fullName>
                            </Person>
                        </authors>
                        <publishingDate> 2001-01-01 </publishingDate>
                        <documentTitle> Blueprints for This Sensor </documentTitle>
                        <documentNumber> YSI-12345-678 </documentNumber>
                    </Reference>
                </supportingDocuments>
            </SensorCreation>
            <SensorDeployment>
                <byWhom>
                    <Person>
                        <fullName>SomeOther Guy</fullName>
                    </Person>
                </byWhom>
                <where> Someplace, Somewhere </where>
                <when> 2001-12-04 </when>
                <description> Attached to northern most footing on Some Bridge at a depth
                    of 5 meters </description>
            </SensorDeployment>
        </history>
        <properties>
            <genericProperty name="sensorTechnology" dataType="xs:string"> rapid
                pulse </genericProperty>
            <genericProperty name="measurementMethod" dataType="xs:string"> EPA
                accepted </genericProperty>
            <genericProperty name="membraneThickness" dataType="xs:double"
                uom="#mil"> 1.0 </genericProperty>
            <genericProperty name="membraneType" dataType="xs:string"> Teflon
                </genericProperty>
            <genericProperty name="probeSolutionType" dataType="xs:string"> Na2SO4
                </genericProperty>
        </properties>
    </SensorSpecification>
</describedBy>
```

Document Metadata



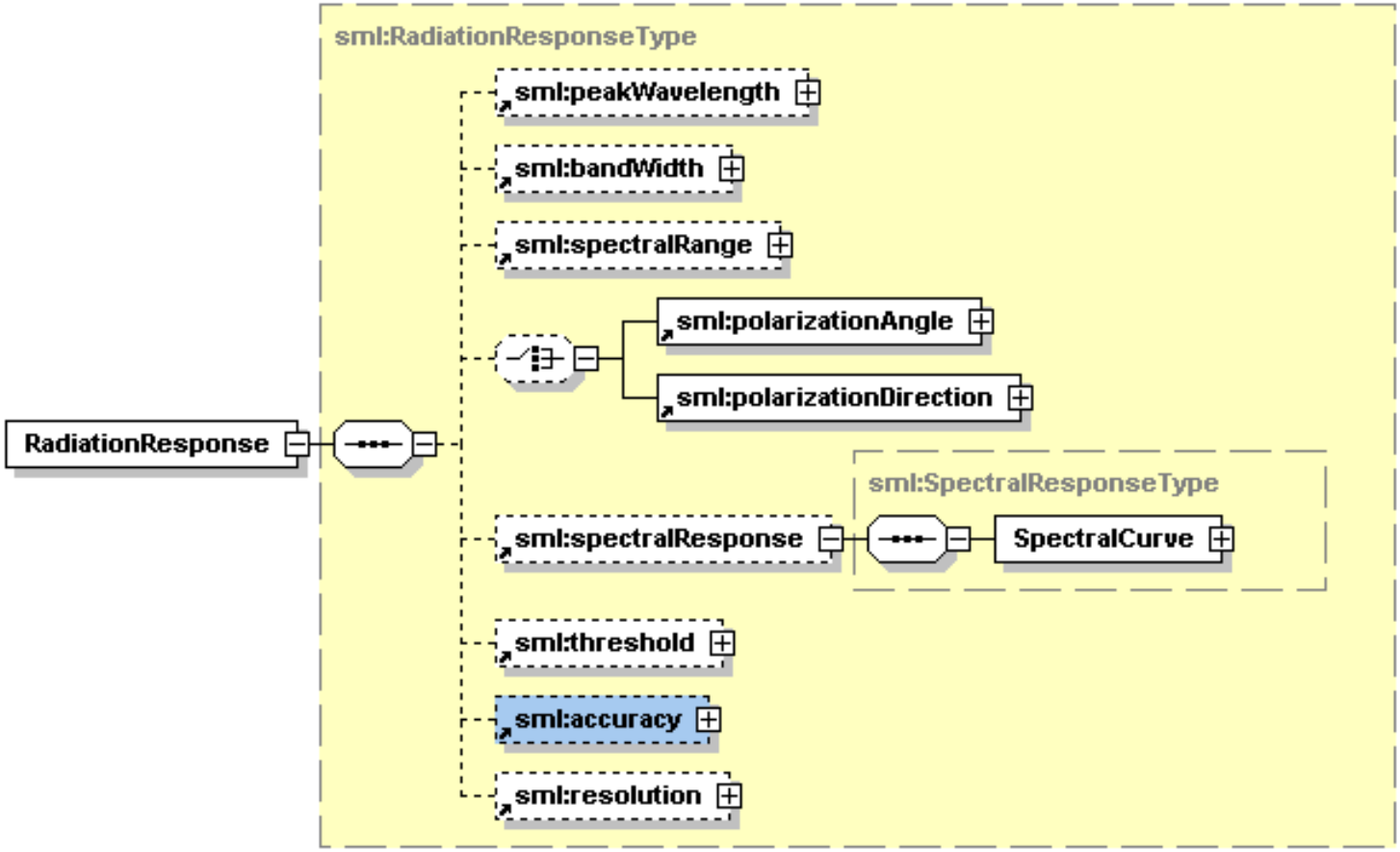
Sensor Response






```
<response id=ysi_wss_0001>
    <dynamicRange>
        <minimum>
            <Quantity observableType=#windSpeed unitOfMeasure=#mph> 0 </Quantity>
        </minimum>
        <maximum>
            <Quantity observableType=#windSpeed unitOfMeasure=#mph> 134 </Quantity>
        </maximum>
    </dynamicRange>
    <threshold>
        <Quantity observableType=#windSpeed unitOfMeasure=#mph> 2.2 </Quantity>
    </threshold>
    <survivableRange>
        <maximum>
            <Quantity observableType=#windSpeed unitOfMeasure=#mph> 220 </Quantity>
        </maximum>
    </survivableRange>
    <operationalRange>
        <minimum>
            <Quantity observableType=#airTemperature unitOfMeasure=#celsius> -40
            </Quantity>
        </minimum>
        <maximum>
            <Quantity observableType=#airTemperature unitOfMeasure=#celsius> 40
            </Quantity>
        </maximum>
    </ operationalRange >
</response>
```

Radiation Response



SensorML Status

Ongoing Directions and Activities

Support for Dynamic Remote Sensors

- Original SensorML DTD definition focused entirely on the geometric and dynamic properties of remote sensors (e.g. scanners and imagers)
- Irony: Current SensorML Schema is devoid of support for dynamic remote sensors
  - Result of OpenGIS OWS1 initial focus on in-situ sensors
  - Current version provides more generality and better support for response characteristics (sensitivity, accuracy, operational environment, etc.)
  - Current schema better supports Frames and Sensor Collections which will be used for defining scanning and imaging properties
- Major focus of next two months will be incorporation of previous scanning and frame camera models into new schema

On-Going Activities and Scheduled Deliverables

ON-GOING:

- Redesigning support for remote sensors
  - Redesigning sample geometry and sensor collection geometries using a Frame schema concept
    - e.g. sample geometry frame related to sensor frame through static or dynamic scan geometry definition
    - Sensor frame related to platform frame through static or dynamic mounting specifications
    - Platform frame related to target world frame through dynamic or static coordinate transforms accounting for platform position and orientation
  - Developing sensor application schemas for scanners, frame cameras, and profilers
  - Developing schema for specification of radiation sensitivity characteristics
  - Developing schema for specification of radiation source characteristics

On-Going Activities and Scheduled Deliverables

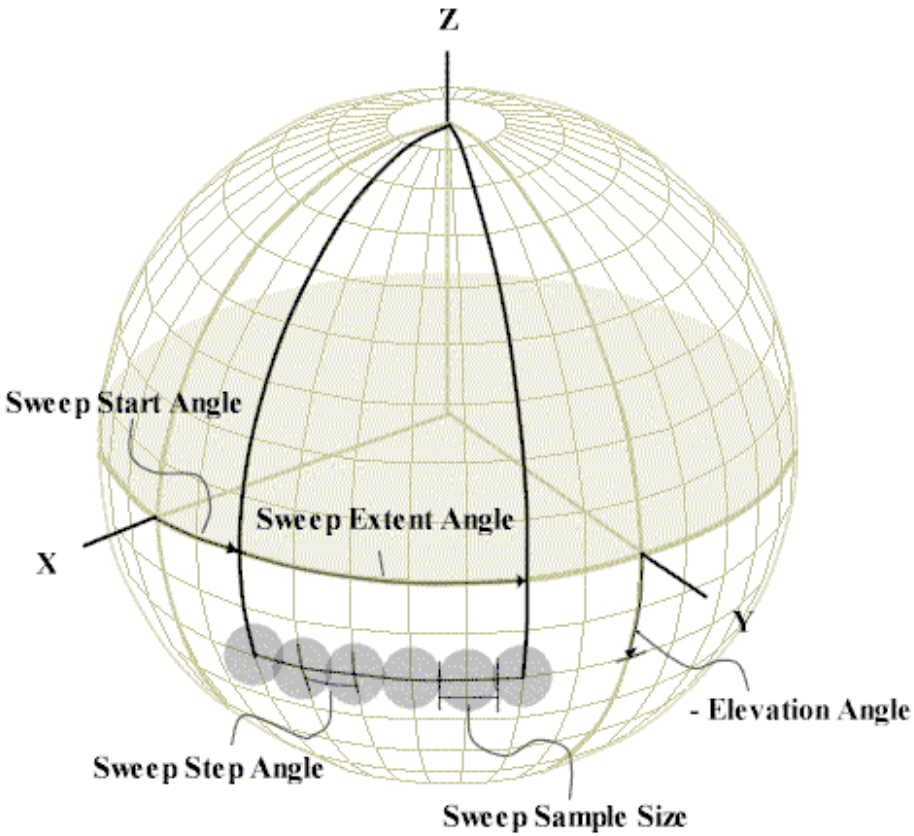
ON-GOING:

- Implement parsers and integrate into Space-Time Toolkit (STT) for testing
- Implement parsers and libraries for processing and geolocating data using SensorML
- Provide and test SensorML documents for wide range of EO sensors
- Provide wizard for creating SensorML documents (12/02)
- Submit SensorML to OpenGIS and ISO TC211 for acceptance

Detailed Near-term Directions

- Support for remote sensors within new schema
  - Geometry and dynamics
  - Bring in concepts previously developed in original SensorML for scanners
  - Design models for frame camera, active sensors, etc.
- Better support for Sensor Collections
- Full support for mobile/dynamic platforms and sensors
- More definitions for Quality and Precision
- More specific ResponseTypes
- Radiation spectral model (used for both radiation source and for instrument sensitivity characteristics)
- More complete examples!

SensorML: Fundamental Geometry Model for Remote Scanners



SensorML Scanning Pricipals

- Geometry and dynamics of scanners and profilers defined in terms of Sweep, Elevation, and Profile coordinates

- Sweep is an angle parameter and is always about the Z (polar) axis,
  Elevation is an angle parameter and about the Y axis,
  Profile is a distance parameter measured from the sensor center

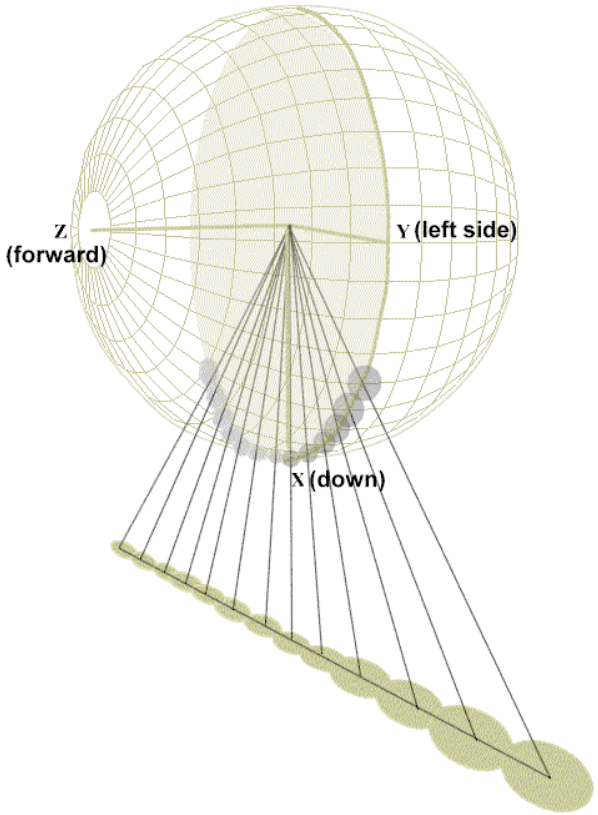
- The “nesting” of Sweep, Elevation, and Profile elements defines the order of scanning

- Conic and linear sensors can be distinguished simply by the orientation of the sensor “spheroid” relative to the target

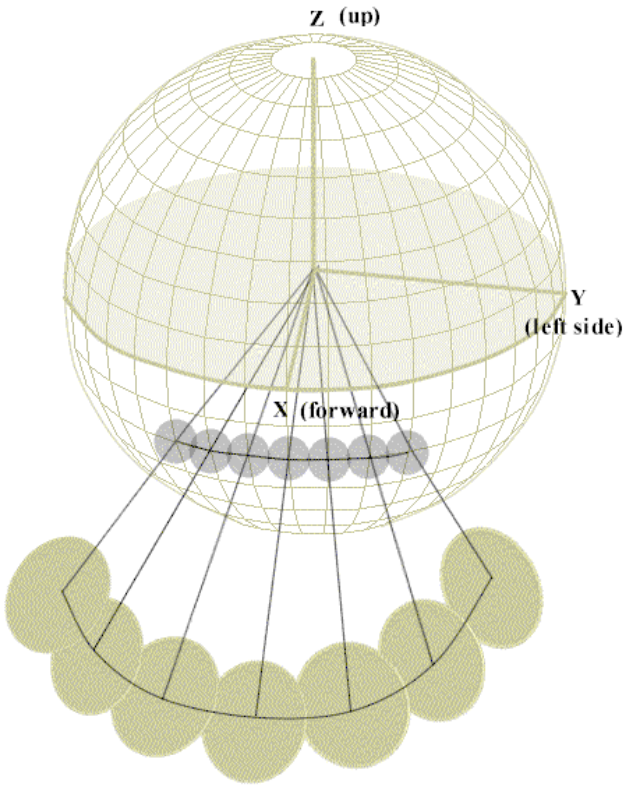
Basic Properties of Scan Geometry/Dynamics

Element	Attributes
SWEEP/ELEVATION	direction, axis
fixedAngle	angleUnits, basis, value
startAngle	angleUnits, basis, value
extentAngle	angleUnits, basis, value
stepAngle	angleUnits, expression, value
startTime	timeUnits, basis, value
extentTime	timeUnits, basis, value
stepTime	timeUnits
repeatTime	timeUnits
numberOfSamples	
PROFILE	direction
fixedDistance	distanceUnits, basis, value
startAngle	distanceUnits, basis, value
extentAngle	distanceUnits, basis, value
stepAngle	distanceUnits, expression, value
numberOfSamples	

Geometry for Different Scanners



Line Scanner



CONIC SCANNER

Transducer Mark Up Language (TML)

Fusion Enabling Technology

Transducer Markup Language (TML)

- **US Government Owned Non-Proprietary**
- Is Software; “Loaded” on or Near Sensor
- Exchanges Raw Sensor Data
- **XML Based**—Fully Compatible with XML
- Normalizes All Data Variables
  - Temporal
  - Spatial
  - Phenomena Values
  - Enables Data for Fusion
- Sensor Agnostic
- Domain Agnostic

Why TML?

What Does TML Do That Other Mark-up Languages Do Not Do?

- Normalizes Data At The Beginning Of The Collection Effort
  - Temporal, Spatial, and Defines Ambiguity
- Sensor Agnostic
- Domain Agnostic
- Net-Centric
- Maintains Native Language (Does Not Corrupt)
- Auditable – Can be Traced Back to Original Data (Not Just Previous Products)

Current Sensor Data

Handling Shortfalls:

- Stove-piped Sensors and Networks
- Enterprise Wide Data Accessibility
- Nonstandard Characterization
- Proprietary
- Nonstandard Data Formats
- “Process” Latent
- Fusion by “Sneaker Net”
- Fusion of Product Data vs. Sensor Data
- Not Scaleable
- Costly

Prevent Knowledge Development and:

Actionable Intelligence
Horizontal Integration
Situational Awareness
Threat Warning
Net Centric Access
Analysis
Counter-CC&D
Combat Effectiveness

Operational View of Sensor Data Management

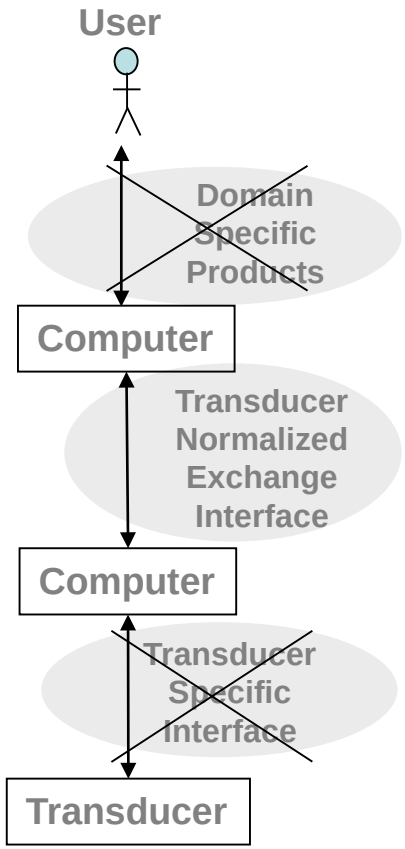
- Sensor Data Exchange
  - Yesterday
  - Today

  } ***A Legacy of Patchwork Workarounds***

  - Tomorrow
    - **Integrated, Horizontally and Vertically Fused**

What is TML?

- **TML is a language for exchanging streaming transducer command & status data**
  - Between a system of transducers and a transducer processor/service or client
  - Self-describing
  - Based on XML (archive or live data)
- **The language communicates**
  - Streaming Transducer data
  - Transducer metadata
  - Metadata is tightly coupled with data
- **Sensor Metadata characterizes the "What", "When" and "Where" of data**
- **Normalizes data for exchange**
  - Standardized spatial, temporal, phenomenon value, and UOM.
  - Data traceable to an enterprise datum's (space, time, value) with uncertainties
  - Data traceability of processing pedigree and source data
- **TML is Sensor Agnostic**
  - Metadata is Common for all Type Sensors
  - Enables a "Common Sensor Processor"
- **TML is Application Domain Agnostic**
- **TML is for Machine-to-Machine data exchange**
  - Historical, live, and future time precision time tagged messages

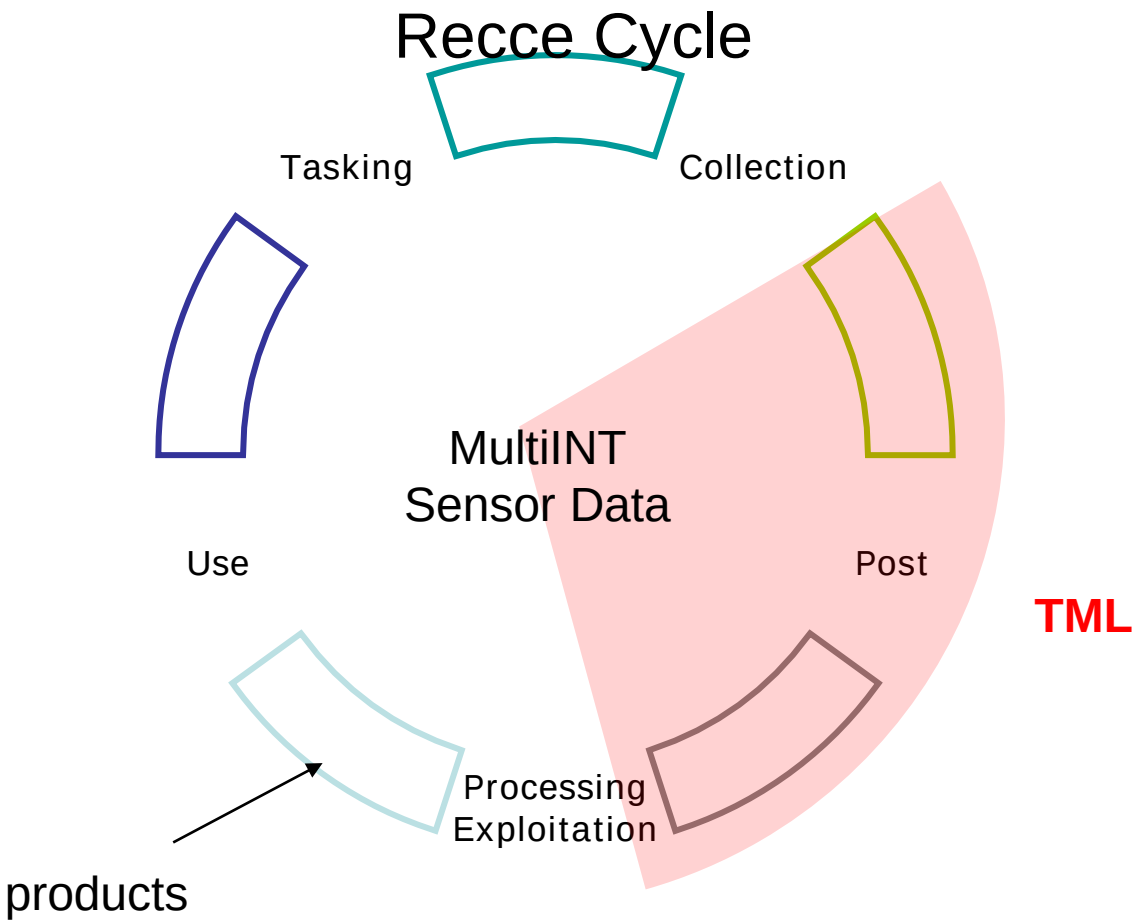


***TML role is not to describe how to represent data**
***Relies on other ontologies and taxonomies to carry domain specific data**
***Relies on other services for communication, security, discovery, delivery, representation, etc.**

TML Enables

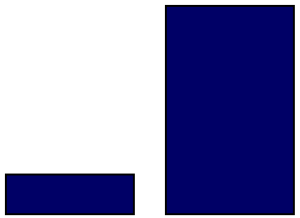
- **Common Processing Environment:**
  - Common Sensor Model
  - Consistent processing for any sensor
  - Sensor Data Fusion
    - Correlation
    - Registration
    - Association of Streaming Multi-Transducer Data
  - Common software tools
- **Interoperable Data Exchange:**
  - Multi-INT, Multi-Domain
  - Scalable
    - Simple to Complex
  - Accurate
    - Precision Geo-Positioning with Error Propagation
  - Efficient
    - Minimal Overhead
  - Live streams and archive files from sensors
  - Live sensor control
  - Post-Before-Process

TML Scope



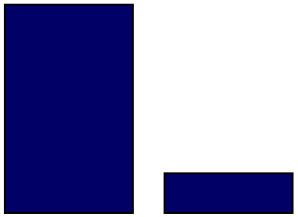
Processing Requirements

Raw Data



Push
Streaming data
Sensor formats

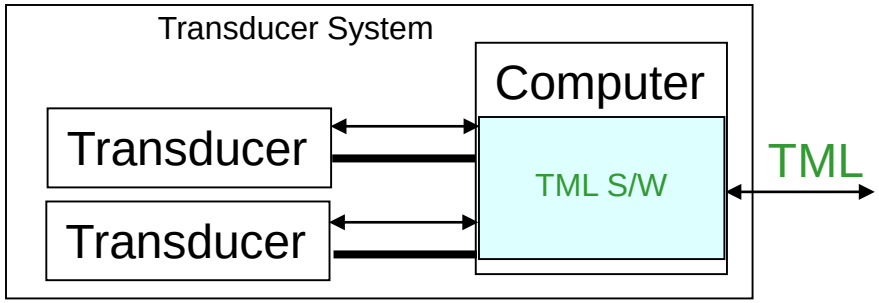
Product Data



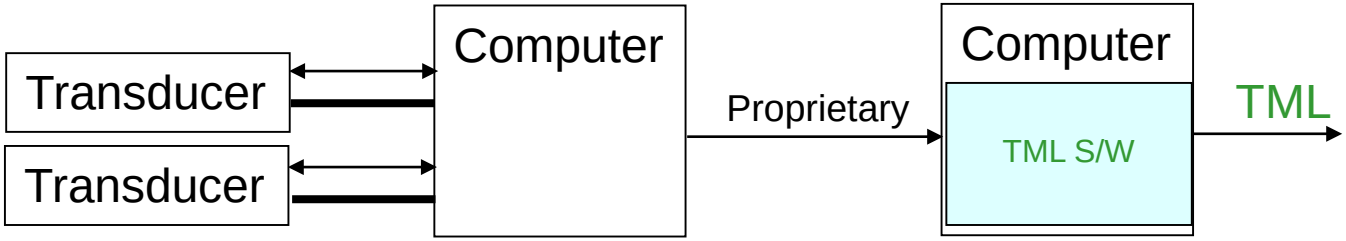
Pull
File data
Display formats

Where is TML Data Born?

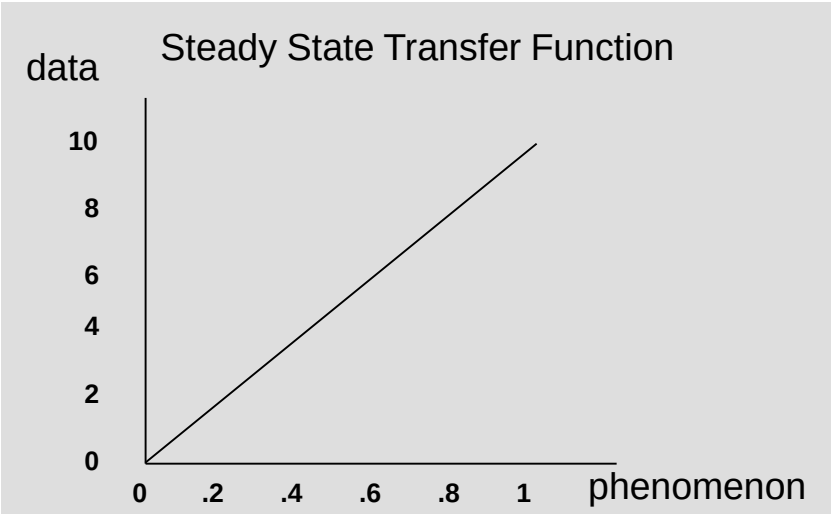
Best at the Sensor

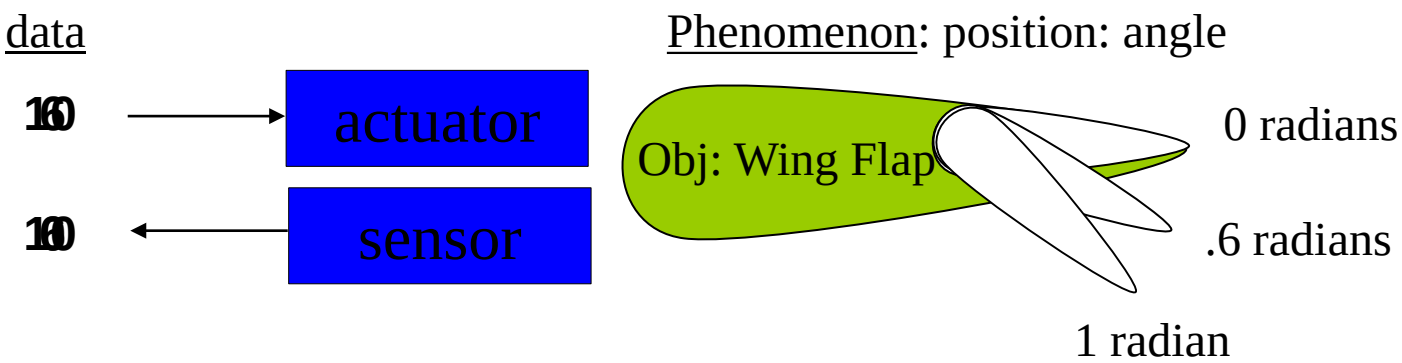


OK downstream

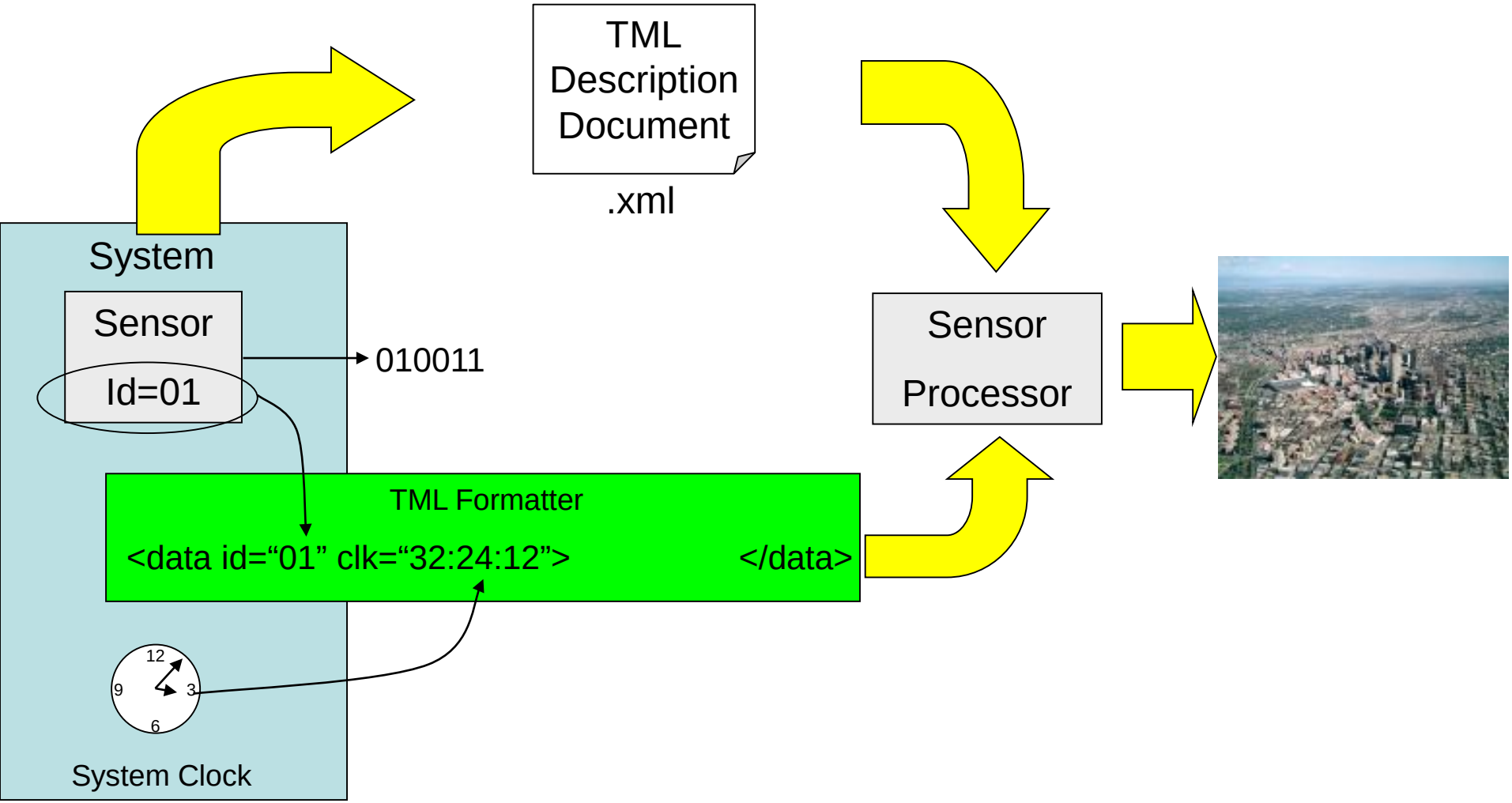


Example: data-phenomenon-object relationship

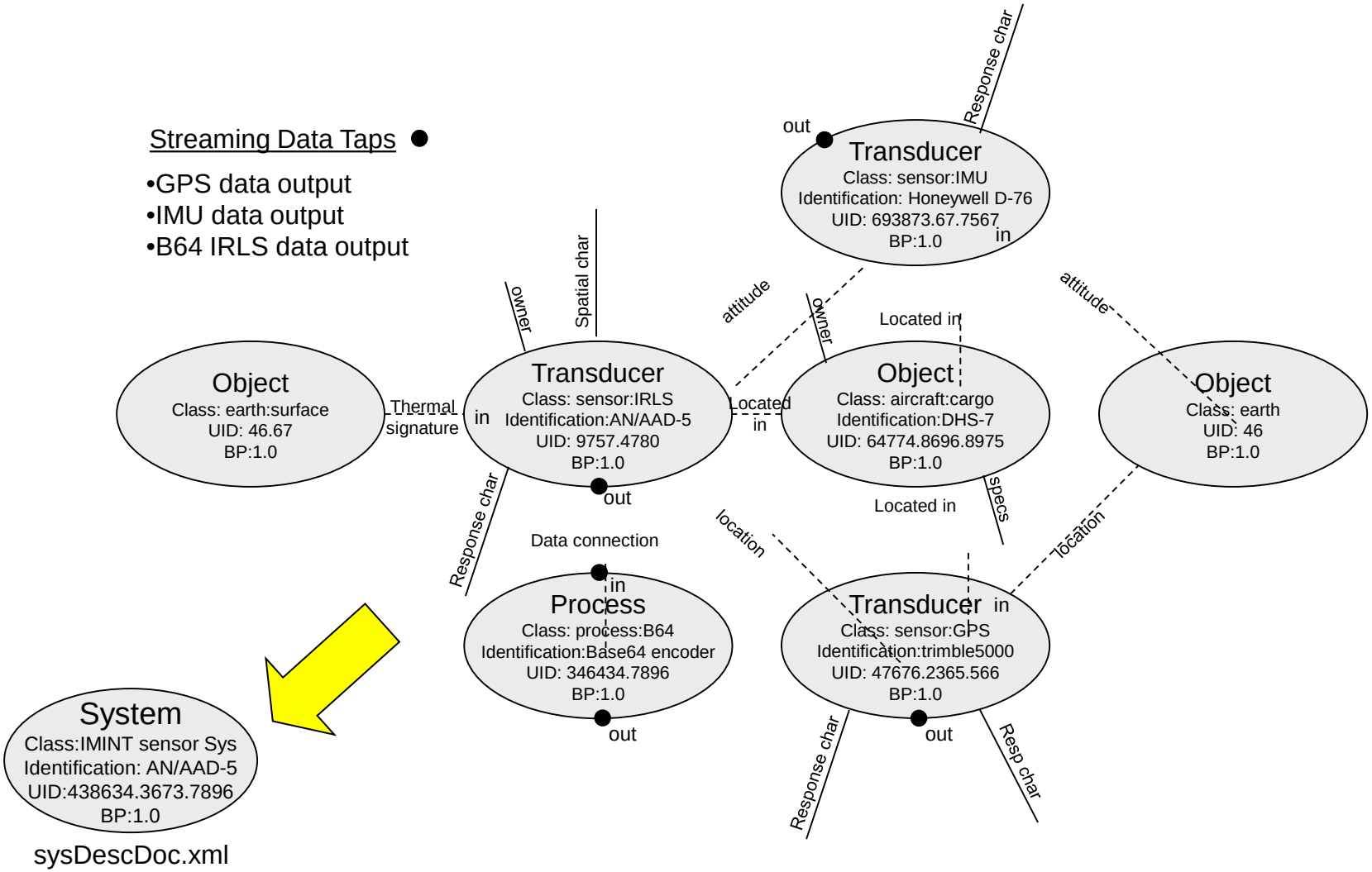




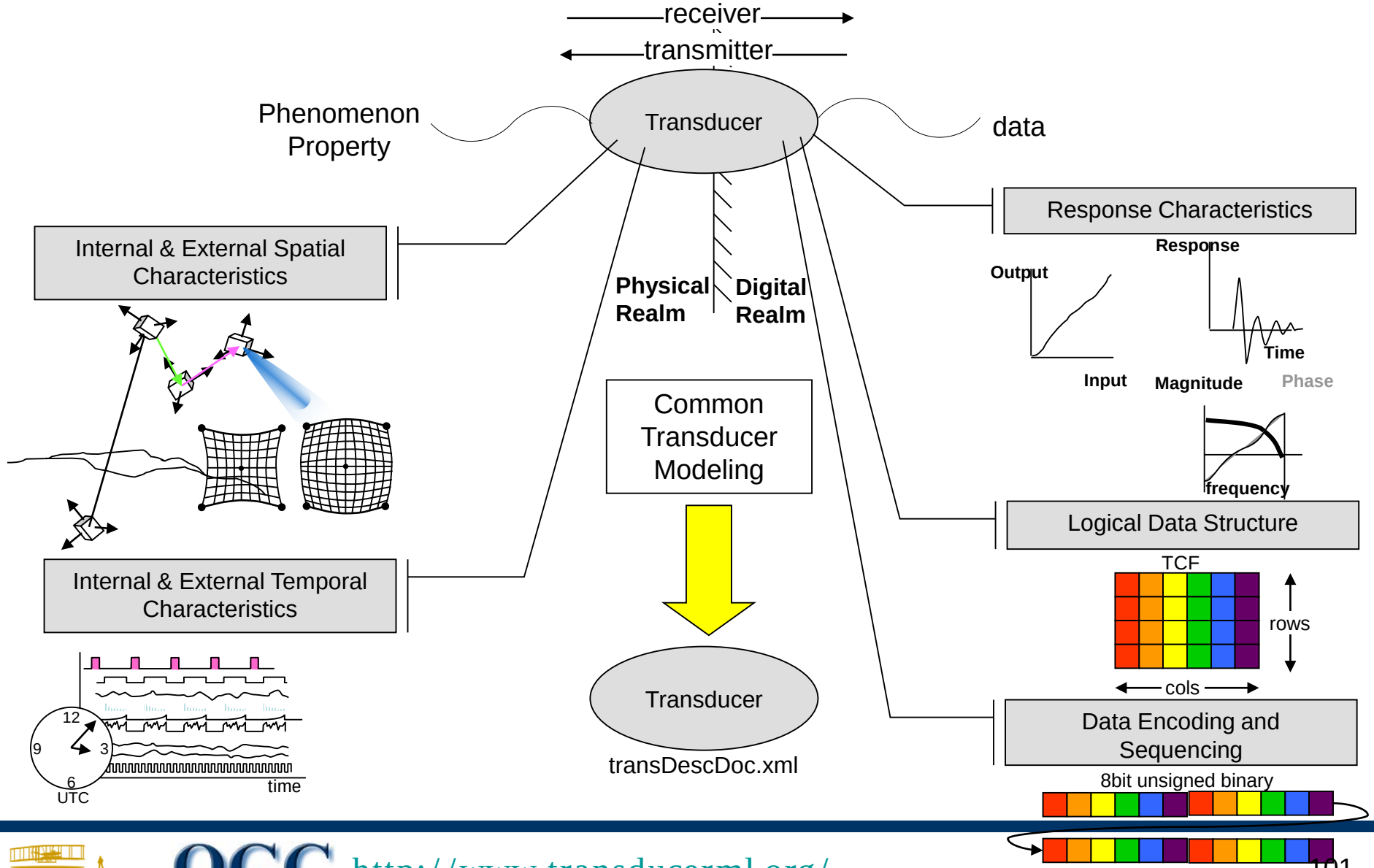
How TML works



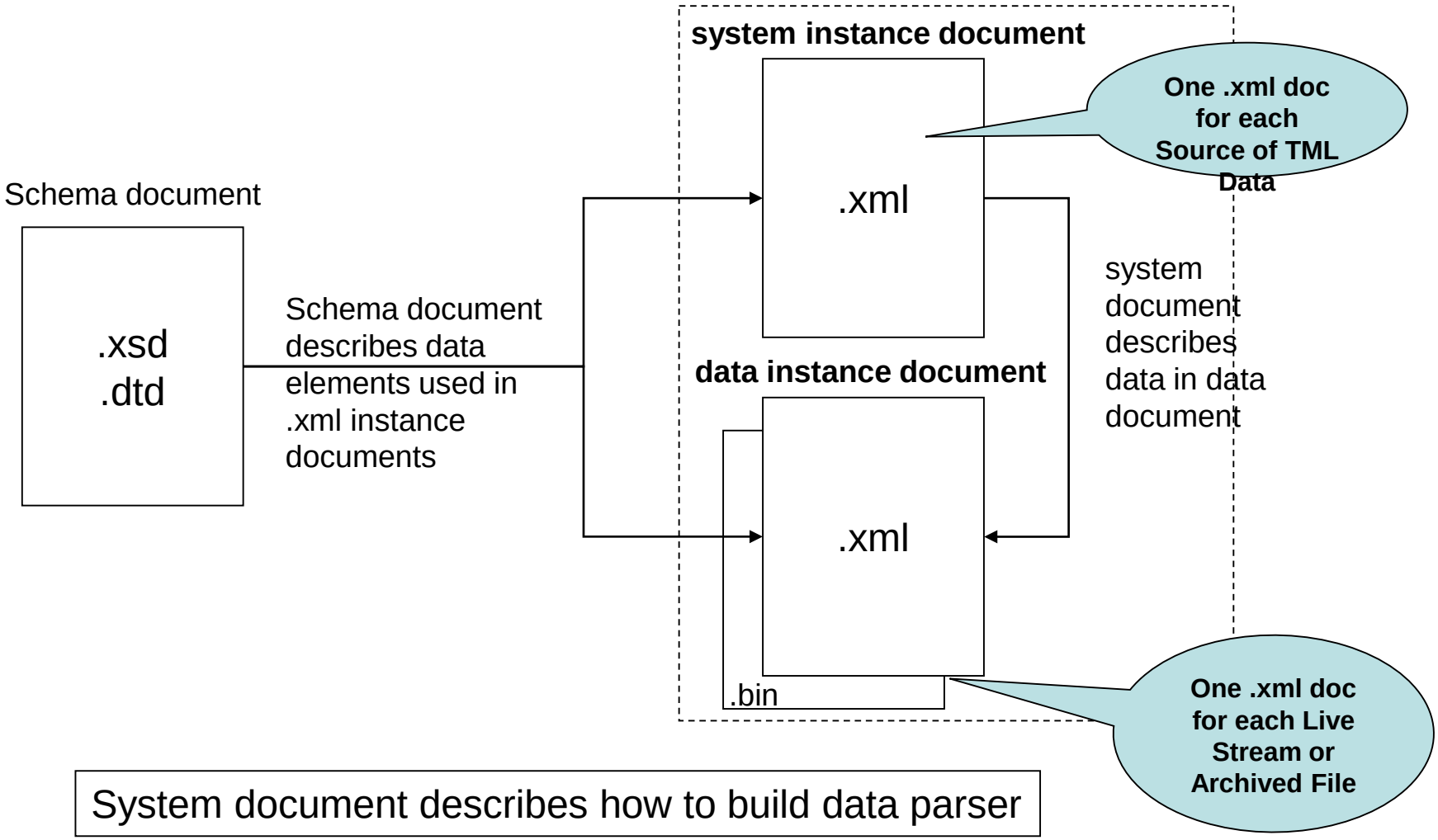
TML System Model



TML Transducer Characterization



TML & XML



System document describes how to build data parser

TML Concept: Static/Dynamic Data

- TML describes the transducer data (Common Transducer Model)

Static data

```
<tml>
    <system>
        <systemClock>…period, count accy</systemClock>
        <transducers>…transducer models…</transducers>
        <process>…process models…</process>
        <relations>…transducer relationships…</relations>
    </system>
    <prodDataDesc>ID mapping,parsing, encoding and sequencing…<prodDataDesc>
<tml>
```

TML Data Stream

Transducer Models
System Topology

- TML transports the transducer data

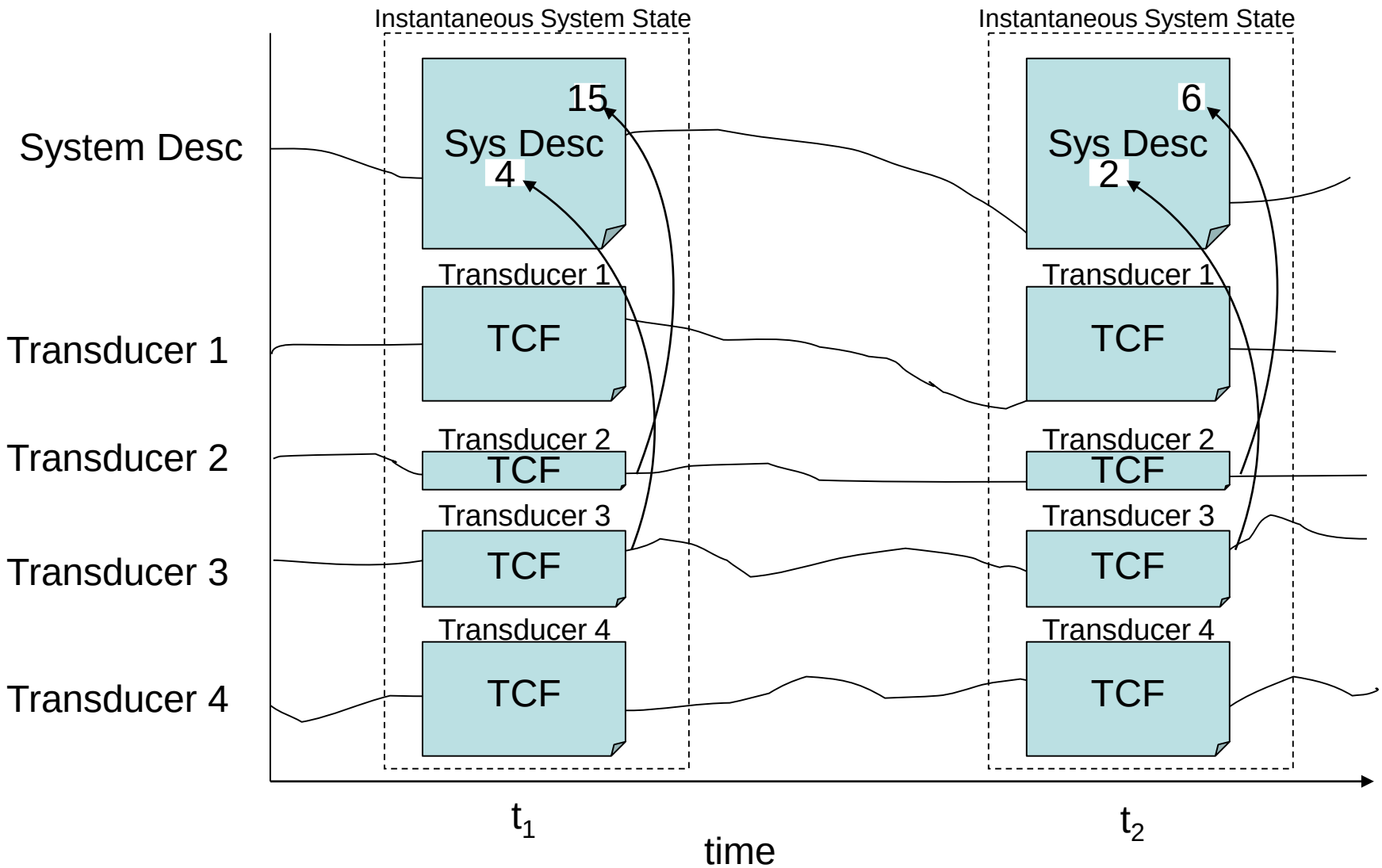
Dynamic data

```
<tml>
    <data ref="t001" clk="3F63B6432674">…transducer data…</ data >
    <data ref="t001" clk="3F63B64326A1">…transducer data…</ data >
    <data ref="t002" clk="3F63B6432701">… transducer data …</ data >
    <data ref="t001" clk="3F63B6432723">… transducer data …</ data >
    <data ref="t003" clk="3F63B643273C">… transducer data …</ data >
    <data ref="t001" clk="3F63B6432767">… transducer data …</ data >
    <data ref="t006" clk="3F63B6432788">… transducer data …</ data >
    <data ref="t001" clk="3F63B64327E9">… transducer data …</ data >
    <data ref="t001" clk="3F63B6432810">… transducer data …</ data >
    <data ref="t001" clk="3F63B6432825">… transducer data …</ data >
    <data ref="t008" clk="3F63B643281B">… transducer data …</ data >
    <data ref="t001" clk="3F63B6432856">… transducer data …</ data >
    <data ref="t002" clk="3F63B6432850">… transducer data …</ data >
<tml>
```

Streaming,
Time Tagged,
Multiplexed,
Transducer Data

time

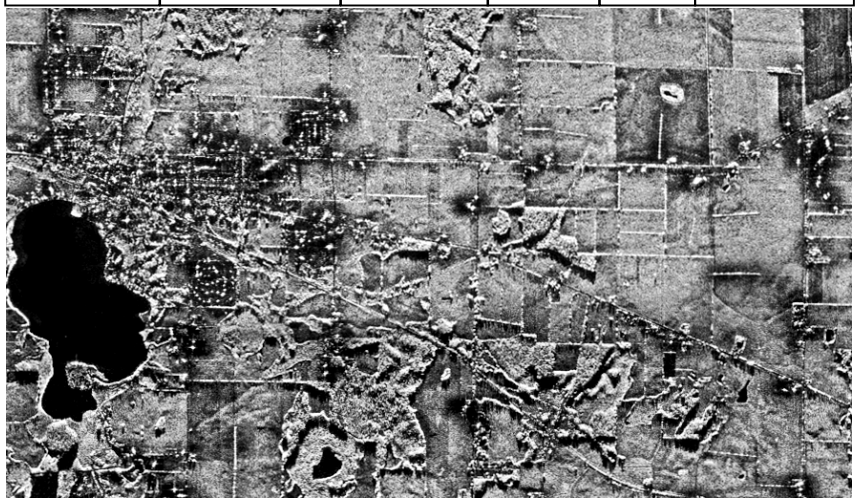
TML Communicates Instantaneous System State



Comparison of Data Structures

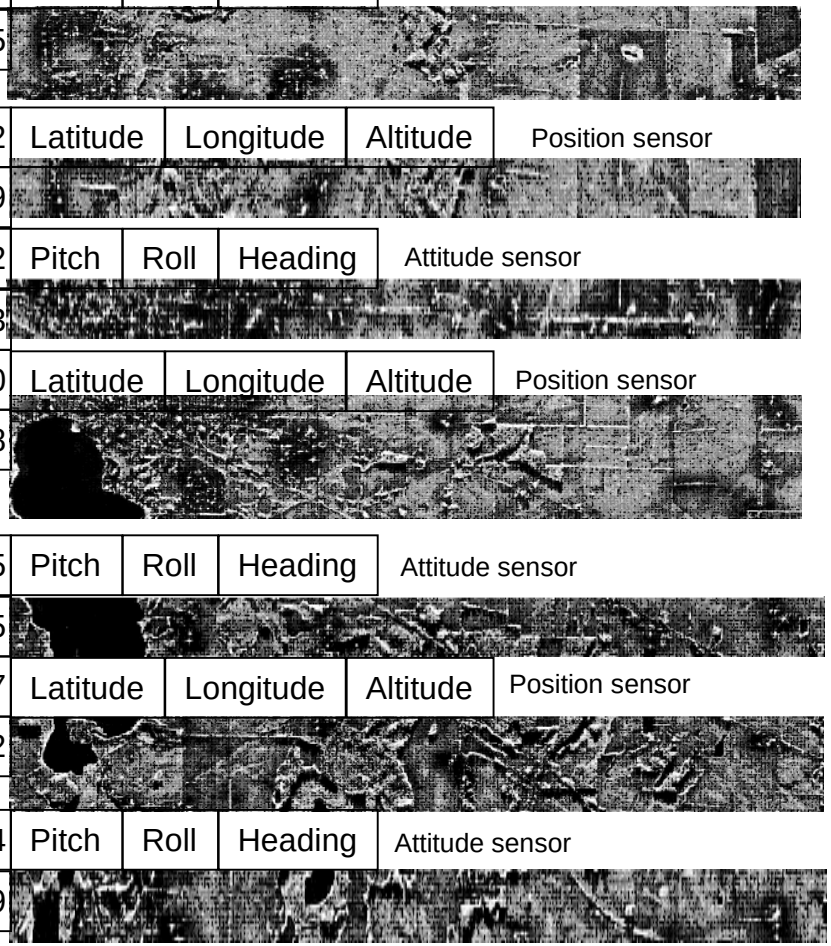
Classical Imagery Format

Latitude	Longitude	Altitude	Pitch	Roll	Heading



TML

t=2456	Pitch	Roll	Heading	Attitude sensor
t=2635				
t=2762	Latitude	Longitude	Altitude	Position sensor
t=2789				
t=2812	Pitch	Roll	Heading	Attitude sensor
t=2893				
t=2910	Latitude	Longitude	Altitude	Position sensor
t=2998				
t=3015	Pitch	Roll	Heading	Attitude sensor
t=3075				
t=3147	Latitude	Longitude	Altitude	Position sensor
t=3222				
t=3354	Pitch	Roll	Heading	Attitude sensor
t=3389				



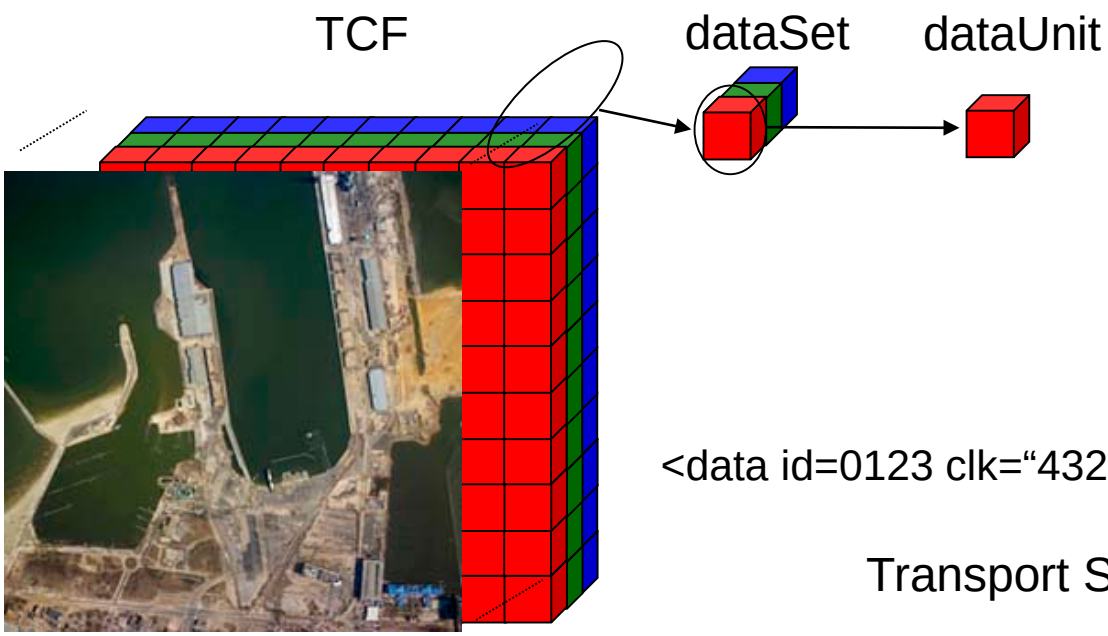
TML Concepts: Logical Data Structure

cluster				
TCF	TCF	TCF	…	TCF

Transducer Characteristic Frame (TCF)				
dataSet	dataSet	dataSet	…	dataSet

dataSet				
dataUnit	dataUnit	dataUnit	…	dataUnit

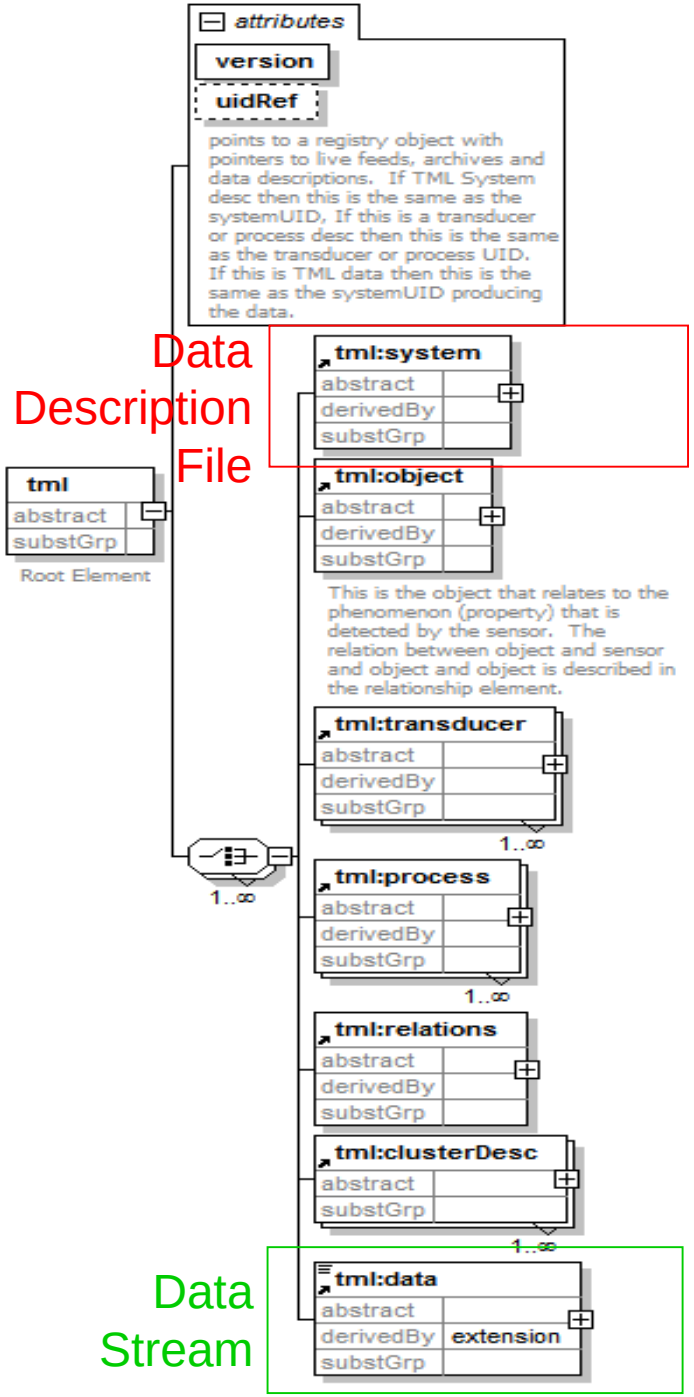
TCF - A logical group of dataSets for grouping data for exchange and associating modeling parameters.

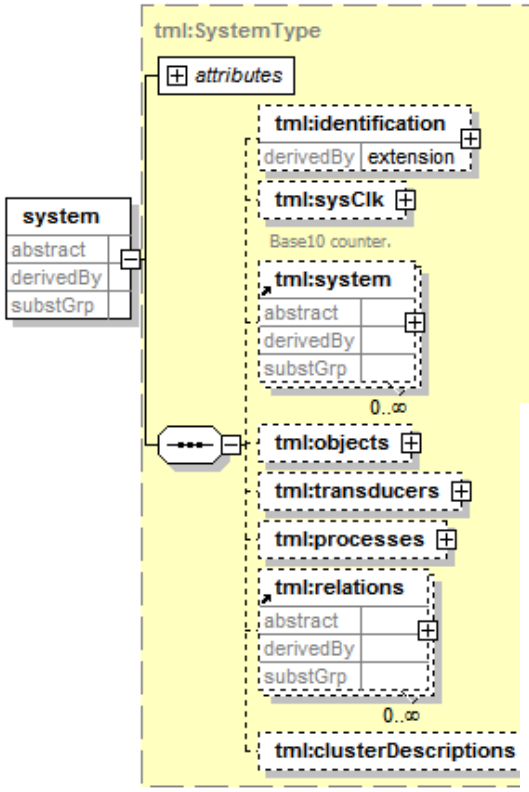


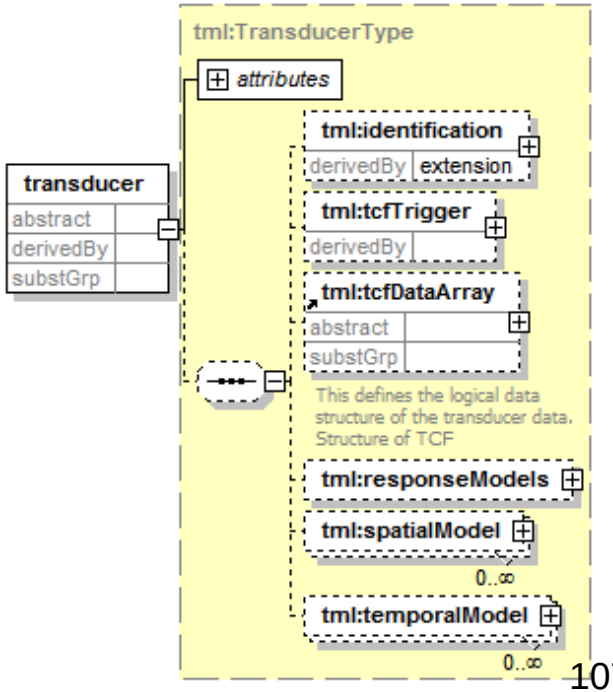
<data id=0123 clk="432654">*…cluster…*</data>

Transport Structure?

TML Framework

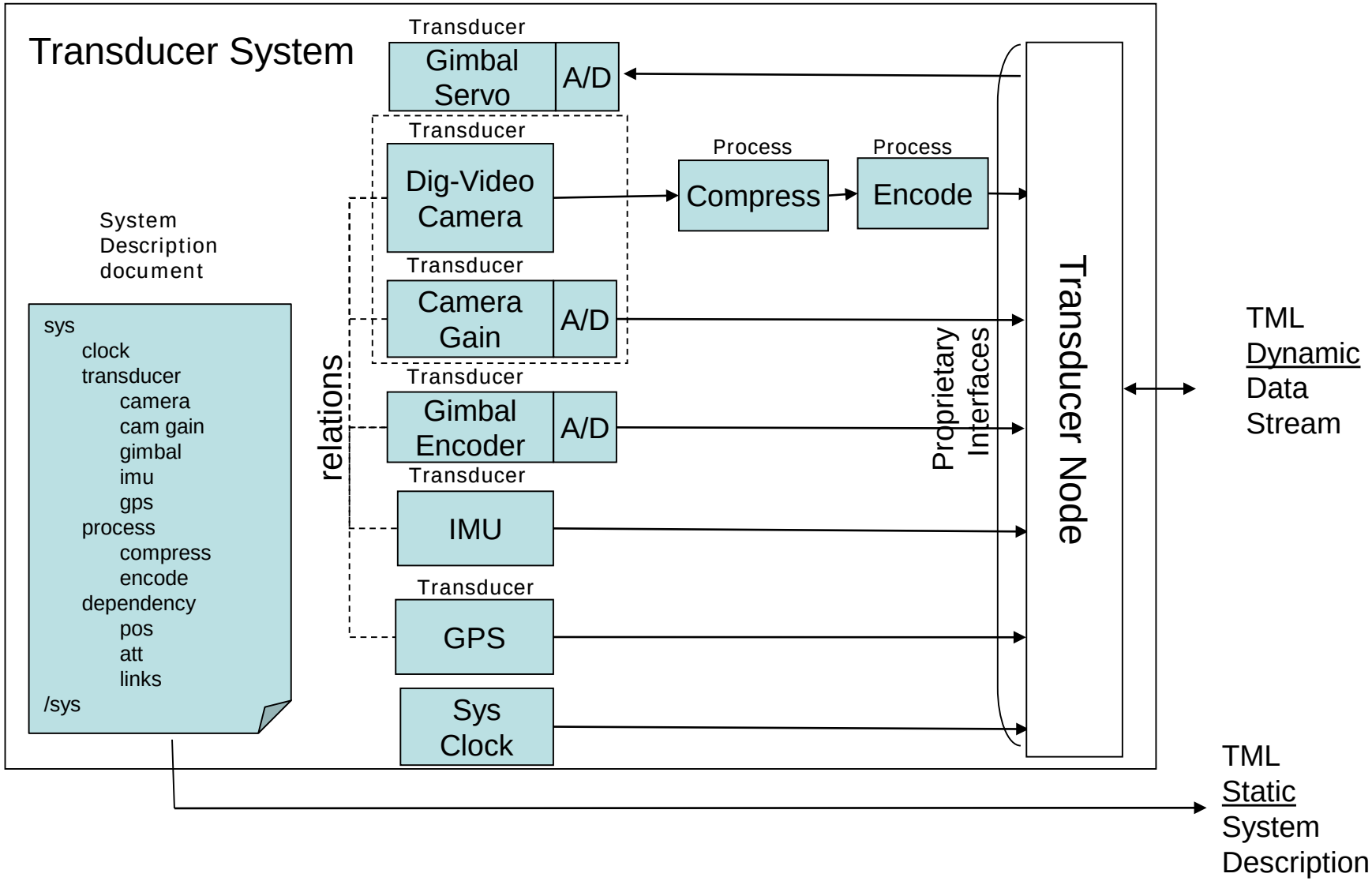






Generated with XMLSpy Schema Editor www.altova.com

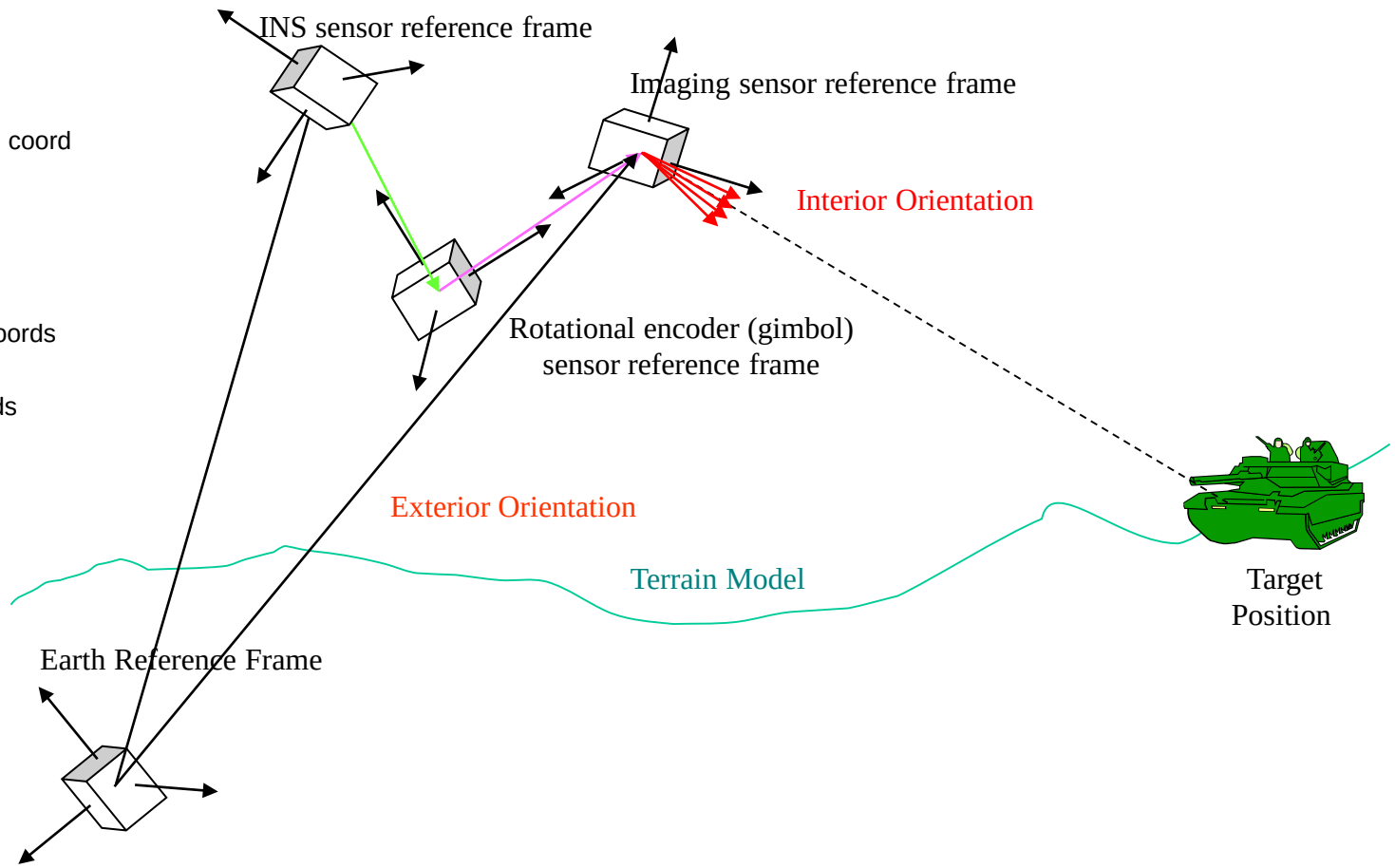
System (transducer | process | relations)



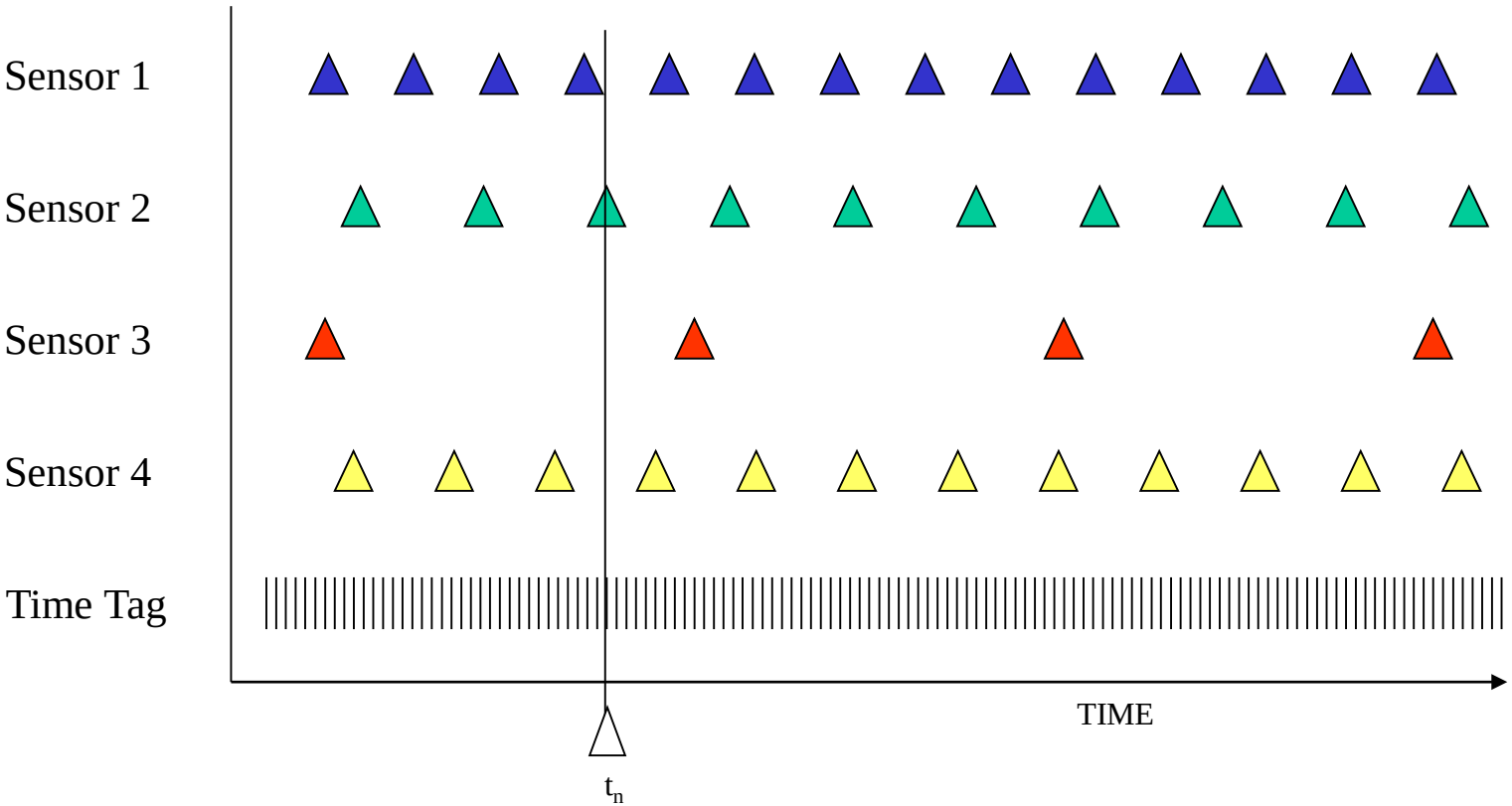
TML Concepts: Relative/Absolute Position

Geometry Characteristics

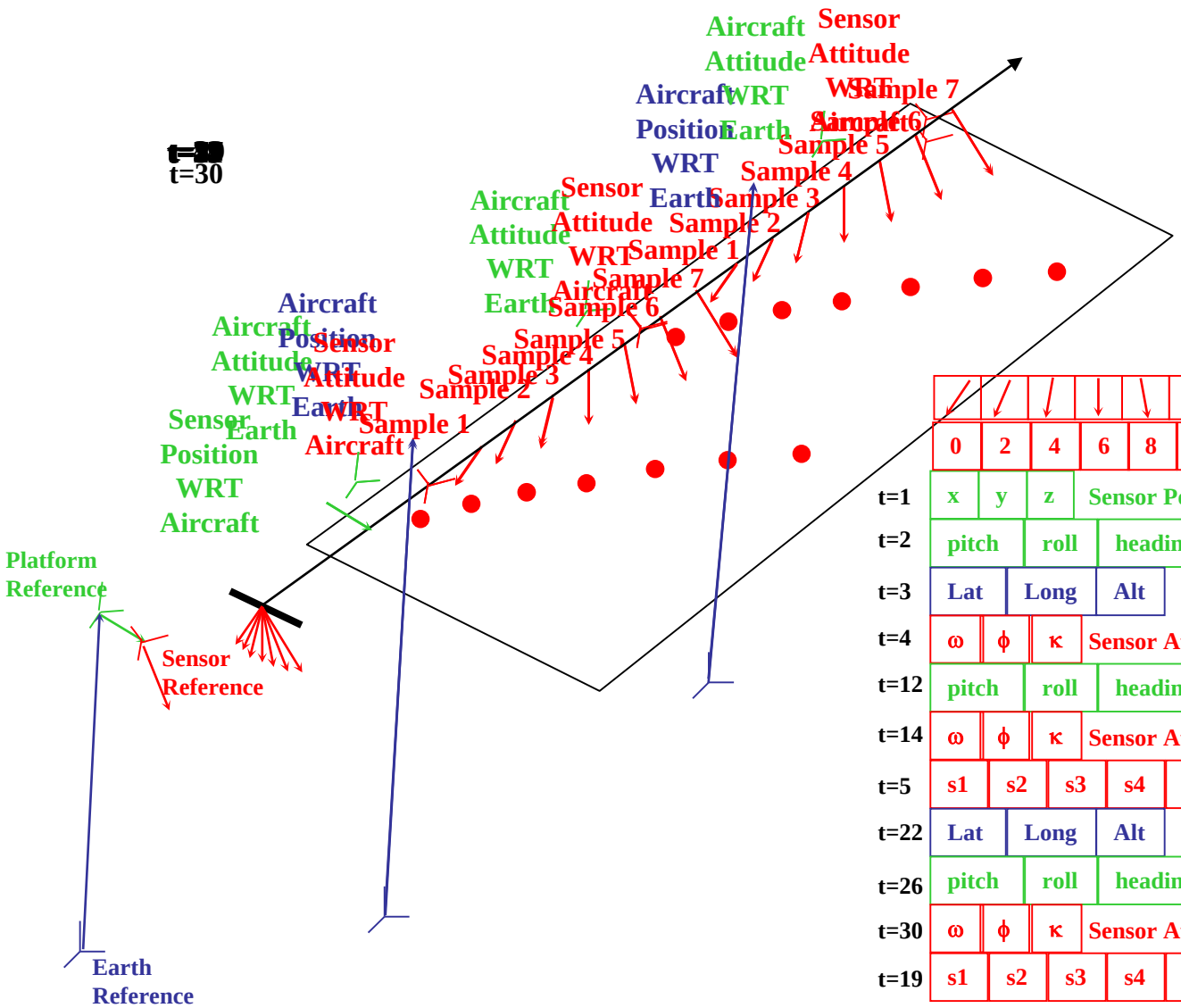
- External
  - Spatial
    - Sensor Position coord
  - Temporal
    - Data – date/time
- Internal
  - Spatial
    - Pixel position coords
  - Temporal
    - Pixel time coords



Asynchronous Sampling in a Multi-sensor System



Time Sequence of Sensor Events

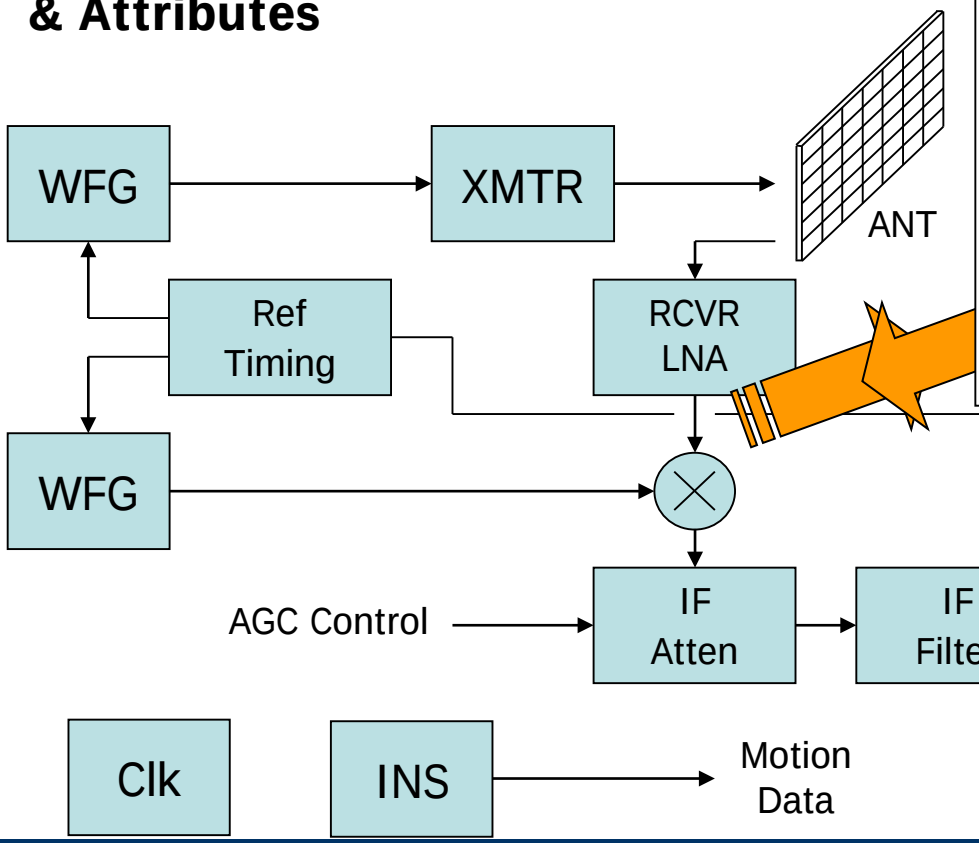


	0	2	4	6	8	10	12	
t=1	x	y	z					Sensor Position WRT GPS
t=2	pitch	roll	heading					Aircraft Attitude WRT Earth
t=3	Lat	Long	Alt					Aircraft Position WRT Earth
t=4	ω	ϕ	κ					Sensor Attitude WRT IMU
t=12	pitch	roll	heading					Aircraft Attitude WRT Earth
t=14	ω	ϕ	κ					Sensor Attitude WRT IMU
t=5	s1	s2	s3	s4	s5	s6	s7	Sensor Frame
t=22	Lat	Long	Alt					Aircraft Position WRT Earth
t=26	pitch	roll	heading					Aircraft Attitude WRT Earth
t=30	ω	ϕ	κ					Sensor Attitude WRT IMU
t=19	s1	s2	s3	s4	s5	s6	s7	Sensor Frame

Sensor model data is only sent once in the beginning

Raw Radar Data Characterization with TML

Characterizes the System & Attributes

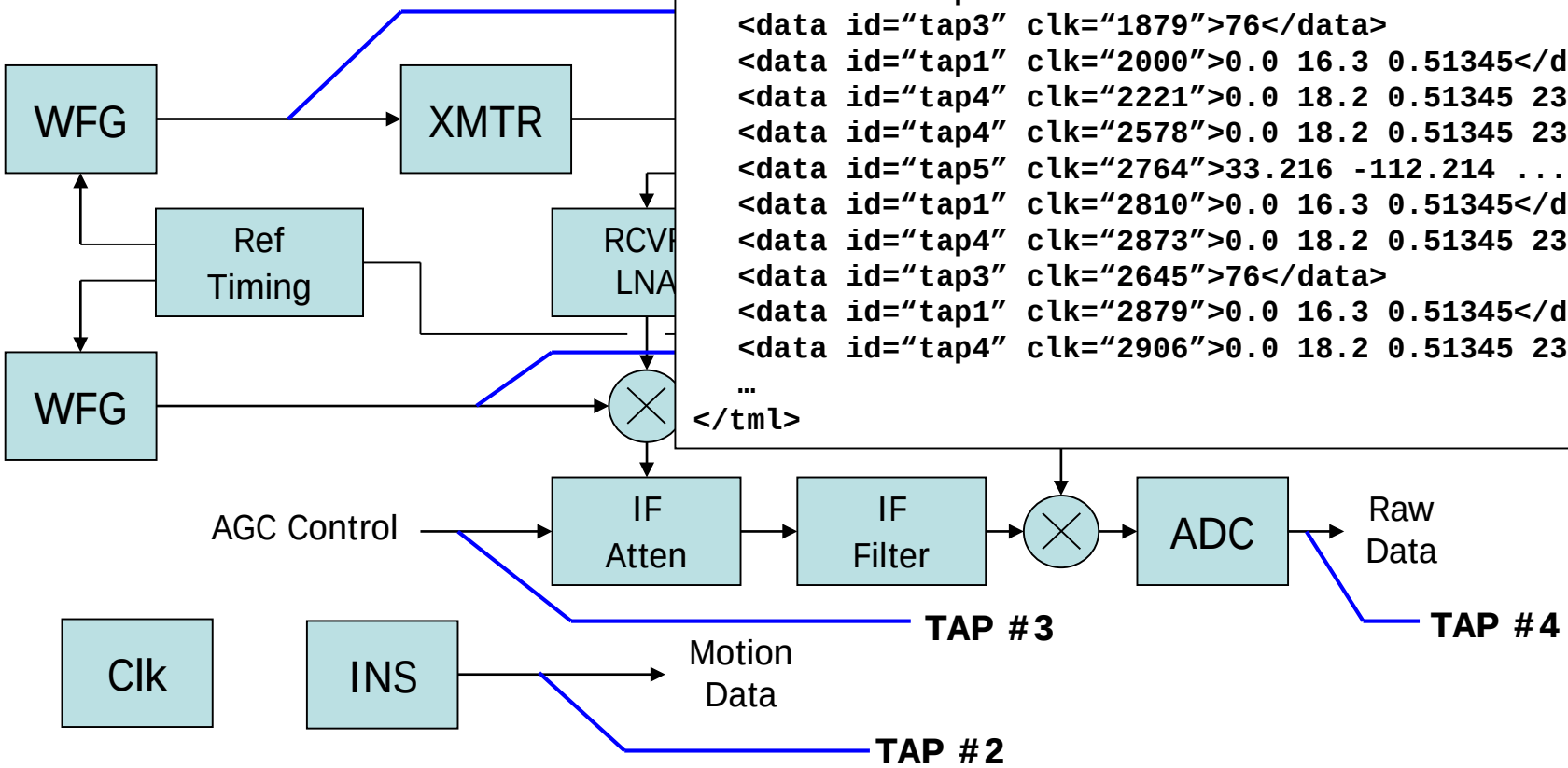



```
<?xml version="1.0" encoding="UTF-8"?>
<tml version="0.92beta 030727">
 <System urn="http://www.tml.org/sys/22"/>
   <transducer type="rec"
urn="http://www.trans.com/r3"/>
   <transducer type="xmtr" urn="http://www.rec.com/t4"/>
   <process name="WFG" urn="http://www.wfg.com/p45"/>
…
   <process name="mixer" urn="http://www.mix.com/p56"/>
   <connect nodeid="1"
      <connect=http://www.wfg.com/p45/out/>
      <connect="http://www.trans.com/t4/in"/>
…
   <connect uid="11"
      <uidRef>http://www.xyz.com/p76/out</uidRef>
      <uidRef>http://www.abc.com/p166/in/uidRef>
   <clusterDesc id="tap1"/>
      …
   <clusterDesc id="tap5"/>
   </System>
</tml>
```

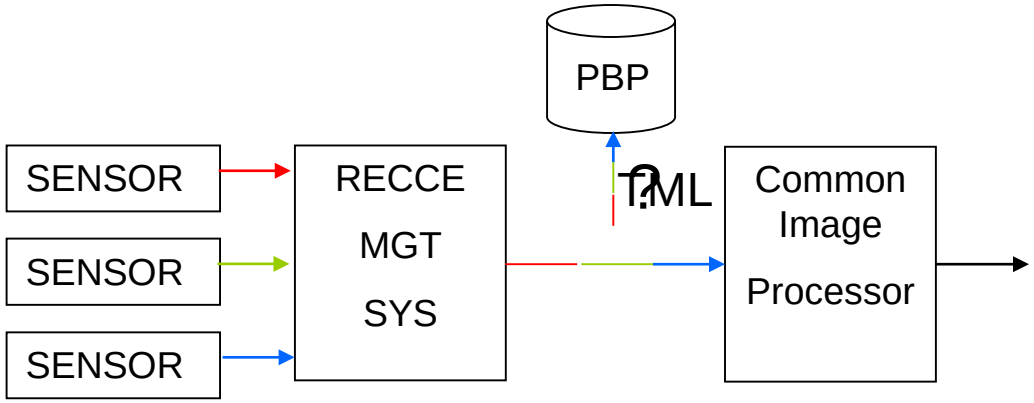
Raw Radar Data Characterization with TML

Characterizes the Dynamic Control and Data Output

```
<tml>
   <system href:xlink="http://www.tml.org/sys/22"/>
   <data id="tap1" clk="0">0.0 16.3 0.51345 200.0</data>
   <data id="tap4" clk="222">0.0 18.2 0.51345 230.0</data>
   <data id="tap5" clk="870">33.216 -112.214 ...  </data>
   <data id="tap1" clk="1000">0.0 16.3 0.51345</data>
   <data id="tap4" clk="1221">0.0 18.2 0.51345 230.0</data>
   <data id="tap3" clk="1879">76</data>
   <data id="tap1" clk="2000">0.0 16.3 0.51345</data>
   <data id="tap4" clk="2221">0.0 18.2 0.51345 230.0</data>
   <data id="tap4" clk="2578">0.0 18.2 0.51345 230.0</data>
   <data id="tap5" clk="2764">33.216 -112.214 ...  </data>
   <data id="tap1" clk="2810">0.0 16.3 0.51345</data>
   <data id="tap4" clk="2873">0.0 18.2 0.51345 230.0</data>
   <data id="tap3" clk="2645">76</data>
   <data id="tap1" clk="2879">0.0 16.3 0.51345</data>
   <data id="tap4" clk="2906">0.0 18.2 0.51345 230.0</data>
   …
</tml>
```

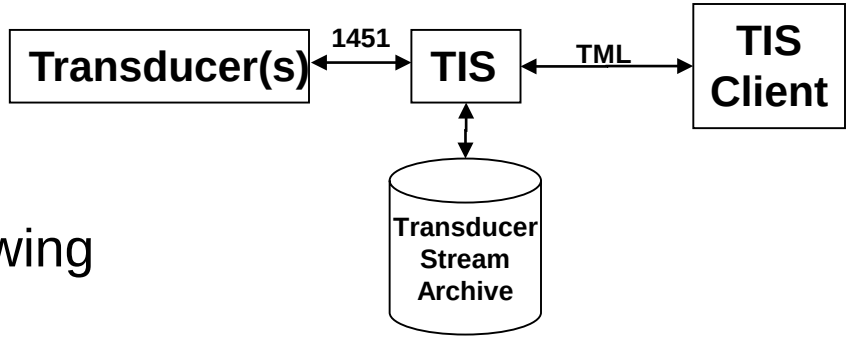


DOD Sensor Data Exchange Formats



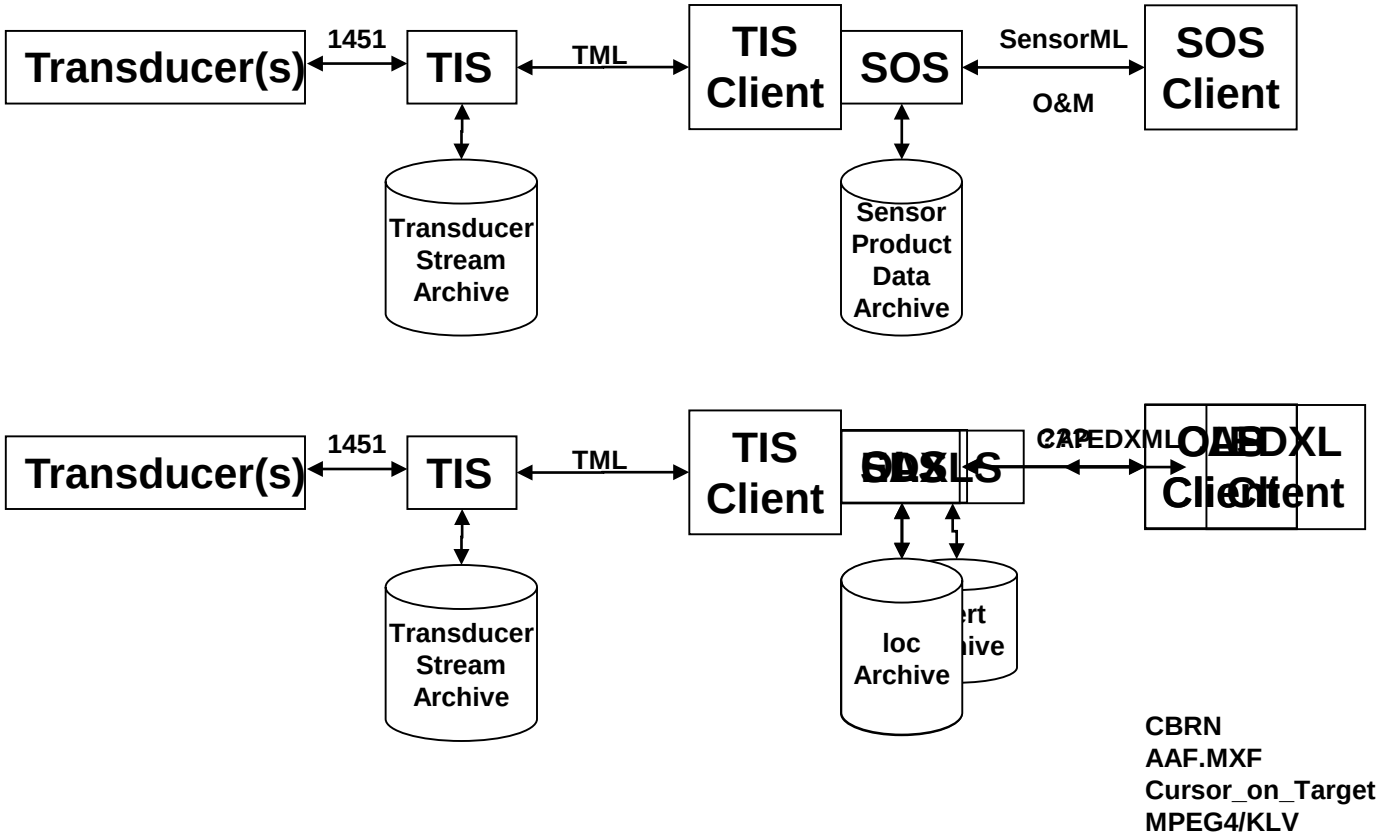
Time Frame	Sensor Interface	Streaming System Data	Sensor Products
Today	Proprietary	Proprietary STANAG 7023 MPEG4/KLV	NITF 2.1 STANAG 4607 STANAG 4609 …
Future	Proprietary IEEE 1451	TML	NITF 3.x/SensorML AAF/MXF/SensorML UPHD/SensorML O&M/SensorML COT…

Transducer Service Functions

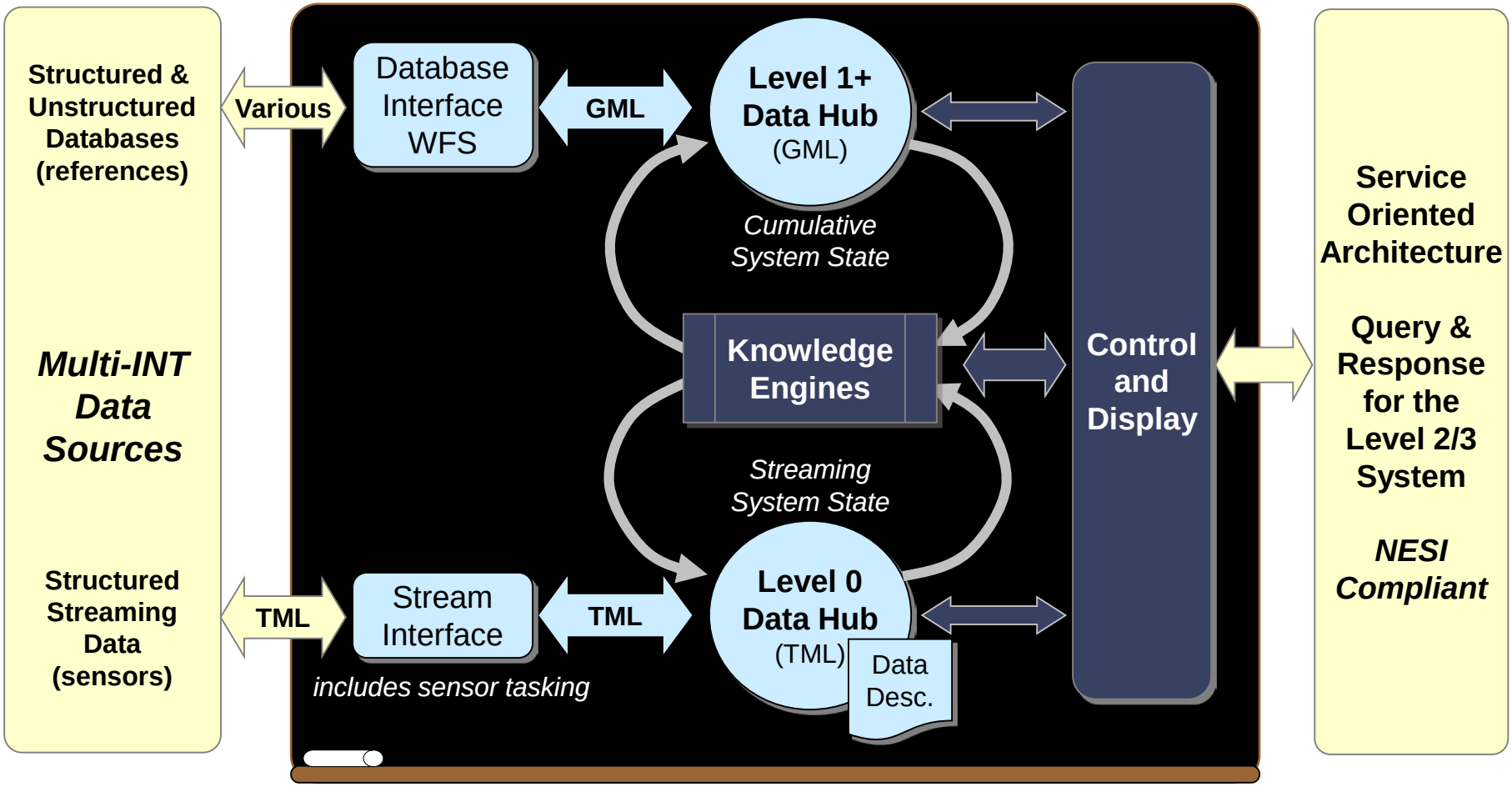


- One or more of the following
  - Live stream connection
    - Data from sensor/point-to-point/broadcast
    - Control to actuator/access control
  - Request live control or outcome from others controlling (SPS)
  - Review control schedule
  - Transducer data description (CS/W)
  - Transducer data archive
  - Transducer archive data catalog (what, when, where)
  - Transducer processing
    - E.g. geo-location, sensor portrayal…
  - Transducer processing description

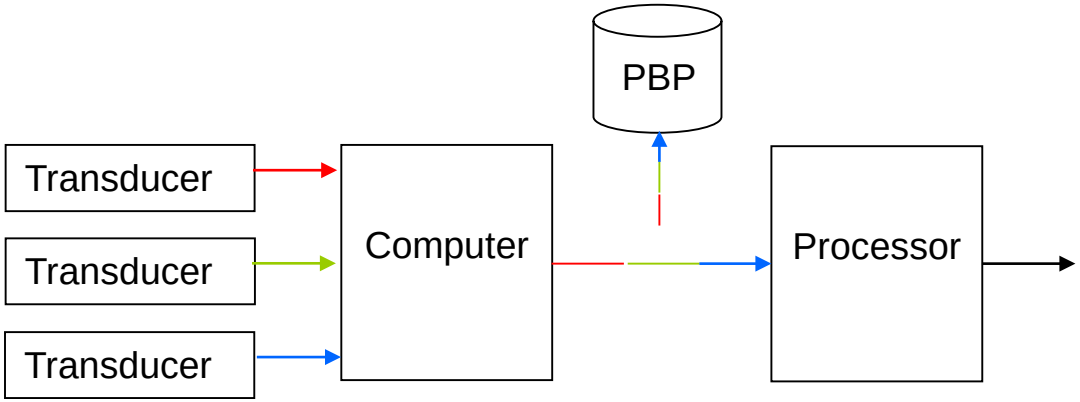
Rationalization of OGC Sensor Standards



Blackboard Data Fusion Service



Harmonization



	1451		TML		SensorML
Data Description	TEDS	↔	System	↔	Process
Data	Data	↔	Data	↔	OM/SWE Common/other

Recap of Presentation

Two current approaches to sensor data management:

1. Application approach
   - GSN - Global Sensor Networks (DERI)
   - Open-source software framework to manage sensor data
   - http://gsn.sourceforge.net
2. Standards approach
   - SWE – Sensor Web Enablement (OGC)
   - Community-standard language framework to manage sensor data
   - http://www.opengeospatial.org/projects/groups/sensorweb

Thank You.

Survey of current

Sensor Network Data Management Frameworks