Existing Standards For Accessing Instruments

Milan Prica
milan.prica@elettra.trieste.it
Sincrotrone Trieste S.C.p.A.

OGF27 - SUMMIT 09
Banff, Alberta, Canada
October 12-15, 2009

Overview

- There are a few solutions to access (scientific) instruments and/or sensors in Grid environment:
  - The first attempt to standardize the access to instruments was the **GTCP** (Globus Teleoperation Control Protocol) by C.Kassleman and L.Pearlman
  - GTCP together with the emerging control systems inspired the design of the **Instrument Element** by the GRIDCC project now maintained by ELETTRA the DORII project
  - A parallel attempt was named **CIMA** (Common Instrument Middleware Architecture) at Indiana University
- OGC standards for describing sensors and measurement processes

OGC: SensorML

- **SensorML** provides standard models and an **XML** encoding for describing any process
- Processes described in SensorML are discoverable and executable
- All processes define their inputs, outputs, parameters, and method, as well as provide relevant metadata
- SensorML models detectors and sensors as processes that convert real phenomena to data
- Note: There is no implementation in code for SensorML as the specification is encoded in XML Schemas.

SensorML Main Concepts 1

- **Component**: Physical atomic process that transforms information from one form to another
- **System**: Composite physically based model of a group or array of components (detectors, actuators, sub-systems)
- **Process Model**: Atomic non-physical processing block usually used within a more complex Process Chain
- **Process Chain**: Composite non-physical processing block consisting of interconnected sub-processes, which can in turn be Process Models or Process Chains

SensorML Main Concepts 2

- **Process Method**: Definition of the behavior and interface of a Process Model
- **Detector**: Atomic component of a composite Measurement System defining sampling and response characteristic of a simple device
- **Sensor**: Specific type of System representing a complete Sensor

Sensor Observation Service

- Part of OGC's **Sensor Web Enablement** (SWE) activity
- Provides an API for managing deployed sensors and retrieving sensor data and specifically “observation” data
- SOS has three mandatory, core operations:
  - GetObservation
  - DescribeSensor
  - GetCapabilities.

CIMA Goals

- Integrate instruments and sensors as real-time data sources into grid computing environments through a Service Oriented Architecture
- Develop a methodology for describing instrument capabilities and functions
- Move production of metadata as close to instruments as possible and facilitate the automatic production of metadata

CIMA Components

- Service architecture:
  - Service code and Instrument-dependent Plug-ins
  - Life-cycle management service
- Instrument description
  - Ontology based, Instrument capabilities, access methods, data
  - Static and dynamic information
- Channel/Parcel protocol
  - REST-like high-level protocol in XML
  - Web Services implementation

CIMA Ontology

- Standardized vocabulary to describe instruments
  - must be extensible, to describe new features, new classes of instruments and sensors,
  - must offer flexibility in machine processing
- Ontology must describe:
  - CIMA instruments and sensors and their physical and logical location,
  - Phenomena detected by instruments/sensors,
  - Communication between instruments and sensors and the CIMA plugins

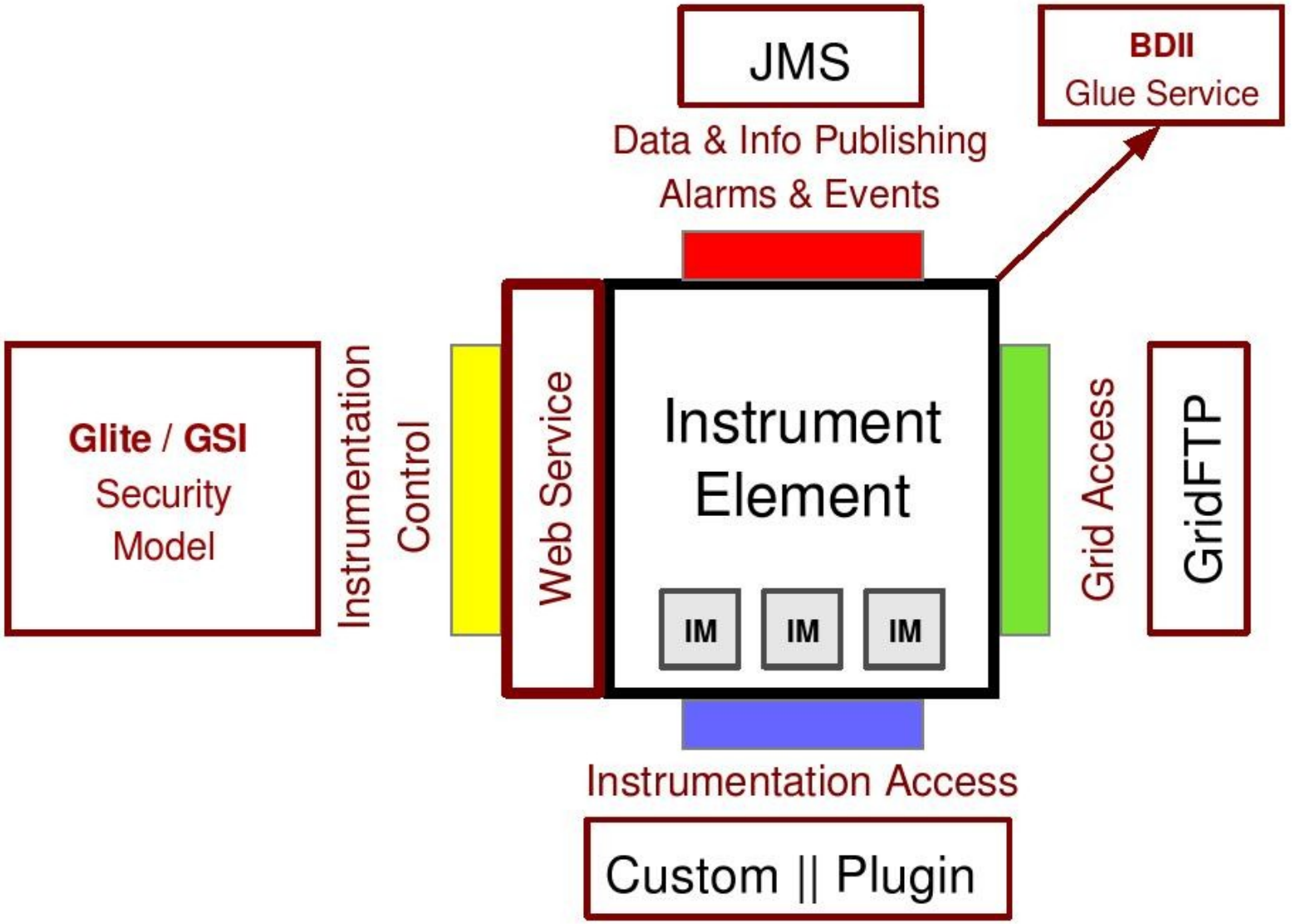
Instrument Element (IE)

- Instrument Element represents **virtualization of data sources**
- **IE provides the traditional Grid with an abstraction of real instruments**, and Grid users with a more interactive interface to control those instruments
- **IE is open source**, pure Java, framework that runs as an Axis WS
- IE provides a flexible solution for connecting a variety of devices to the Grid

IE Features

- Original concept from GRIDCC
- New implementation from ELETTRA, adopted by DORII
- Device centric design
- True multi-user support, concurrent access control
- Grid certificates based authentication and authorization
- Middleware level integration with Grid storage (GridFTP)
- JMS based alarms, events and asynchronous monitoring
- VCR, g-Eclipse, Common Library clients available
- IE is part of the EGEE's RESPECT programme

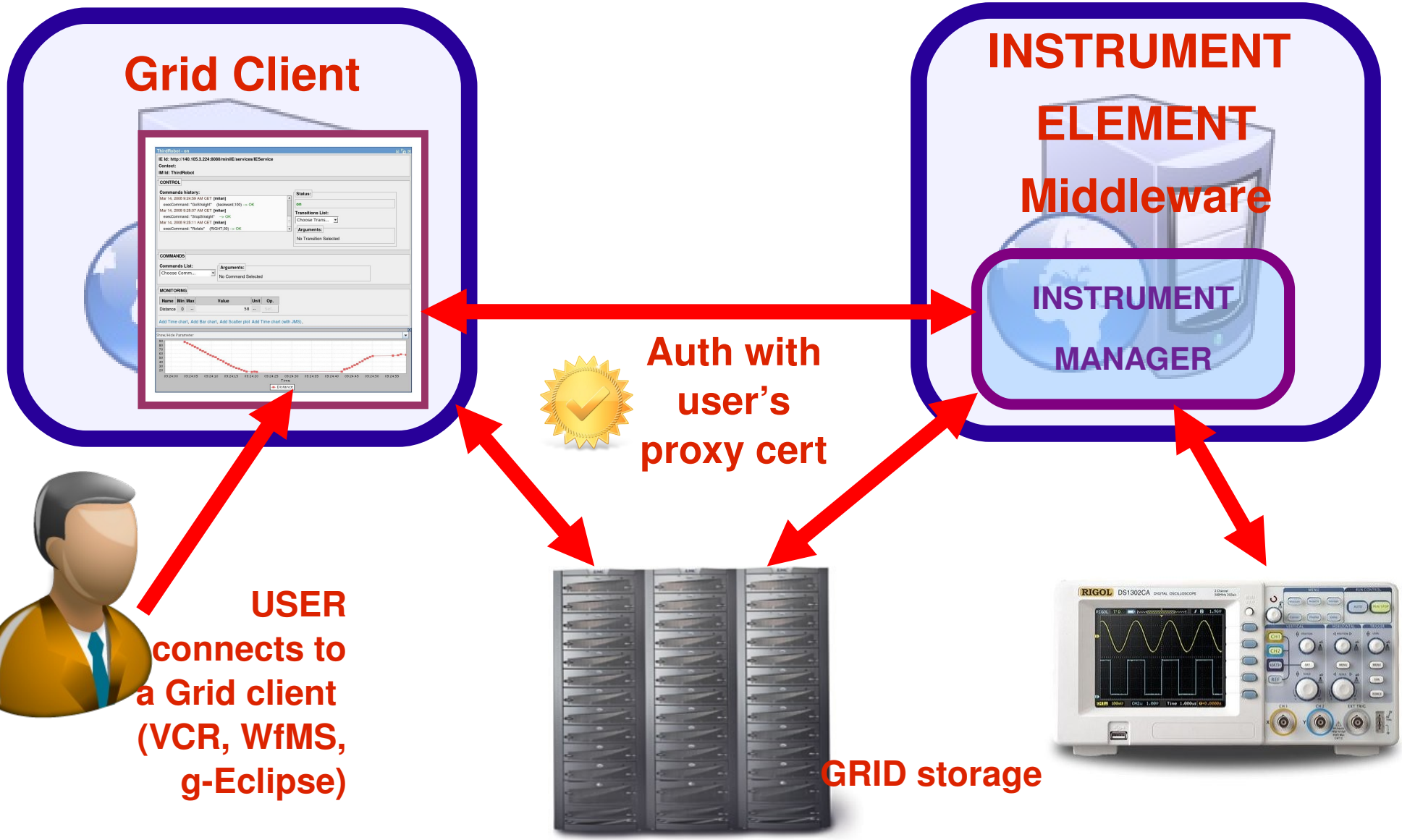
IE Architecture



Instrument Manager

- **Java** client for the Instrument; runs in IE framework
- Fully described by the **XML descriptor**
  - Possible IM states (→ State Machine), details about Commands, Parameters and Attributes.
- Java classes implement Commands (Transactions) Parameters and Attributes
  - **Attributes** are instrument variables
  - **Parameters** regard instrument settings
  - **Commands** are instructions to be passed to the instrument (interface). May include input parameters
  - **Transactions** are commands that trigger state change

From the user's perspective



Conclusions

- OGC standards are very detailed in describing sensors and the collected data but do not address the security issue -> incomplete for use in Grids
- There is a working implementation of the Grid-aware middleware (IE) for accessing remote instrumentation but it is not a standard proposal
- There is lot of space for defining a new standard for accessing remote instrumentation in Grid environment starting from the above solutions and expanding on them

Thanks for Your attention!

Questions ??

Questions ??