Instrument Elements at Work

- Brunel
- Imperial
- Elettra
- IASA
- IBM
- GRNet
- CNIT
- IMAA
- UniUD
- INFN
- Monitor GUI
The Basic Idea

- Problem Solver
- Security Service
- Information and Monitor Srvc
- Instrument Element
- Virtual Control
- Service Interface
- Execution Service
- Computing Resources
- Storage Resources
- Existing Computational Infrastructures
1: Provides a uniform access to the physical devices

2: Allows a standard access to the instruments

3: Allows the cooperation between different instruments that belong to different VOs
What we are addressing in this demo

We are considering Instruments that are:
1) Large in Number
2) Highly dynamic
3) Widely Distributed
4) In Low Resources/Embedded Systems

Example of Widely Sparse Instrumentation
- Power Grids
- Territory Monitoring
- Sea Monitoring
- Distributed Laboratories
- Transportation Monitoring
- Sensor Network
Widely Sparse Instrument Elements at Work
Instrument Element Discovery process

Instrument Elements

Network Core

Index Services

VCR Monitor GUI

<PeersTable = empty>
1. <PeersTable = 0,A,B,C,D>
2. <peer connection with B>
3. <Look for “M” peers>
4. <direct contact with M2>
4. <Routed contact with M2>
Instrument Element Discovery Process (2)

1. <PeersTable = 0, D, K, X, Y, Z>
2. <peer connection with Y>
3. <Look For “M” peers>
4. <direct contact with Mb>
4. <Routed contact with Ma>
Advantages of a Peer To Peer Approach

- Instruments, Network Core Machines and Network Information Providers can dynamically join and leave the network without affecting other Instruments or requiring additional configuration efforts.
- Instrument Element network will self-optimize. In other words, if you encounter performance problems, you can simply add a “Network Core Machine” and the machines loads will be redistributed.
- The scalability outperforms a Centralized based system.
- In systems based only on an Instrument Element network you could avoid the installation of one or more Centralized services.
- This discovery system can be embedded in IEs that run in FPGA and not in “standard PC”.
- A particular version Instrument Index Service can talk to central index in order to fill the information of a group of “light instruments” that run in embedded systems.
Subscribers subscribe to a given Topic/Queue with a subscribe condition.

Publishers asynchronously publish messages in a Topic/Queue with a given message condition.

Publishers and subscribers can be part of the same WAN distributed machines or not.

Each Instrument can be a data publisher or a data subscriber.

Web Service performance is inadequate for fast publishing.
IMS for Monitoring purposes in this Demo

- Service Oriented Architecture
- Several different WS-I compliant GUI visualizers: J2EE, LabVIEW etc ...
- Adherence to standards: WS-I compliant Web Services
  - Standardized and uniform access to sub-components of the system
  - Services loosely coupled
  - High interoperability

Monitor GUls
IMS Message Rate: one-to-many

- 32 Dual Xeon 2.4GHz 1.5GB RAM machines, 1 GB Ethernet switch
- At most 1 publisher, subscriber, or broker- (Sun MQ3.6) per machine
- No message lost
- RMM-JMS throughput: 75-90 Mbytes/sec. (for 5 and more publishers)