Instruments and Sensors on the Grid: Issues and Challenges

Donald F. McMullen[1],
Randall Bramley[1],
Kenneth Chiu[2],
John Huffman[1],
Tharaka Devidithya[1],
Kia Huffman[1],
Wei Lu[1],
Sameer Tilak[2]

[1]Indiana University

[2]Grid Computing Research Laboratory
SUNY Binghamton

Laura Pearlman
USC/ISI

Marina del Rey, CA

Supported by NSF ANI-0330568.

Motivation

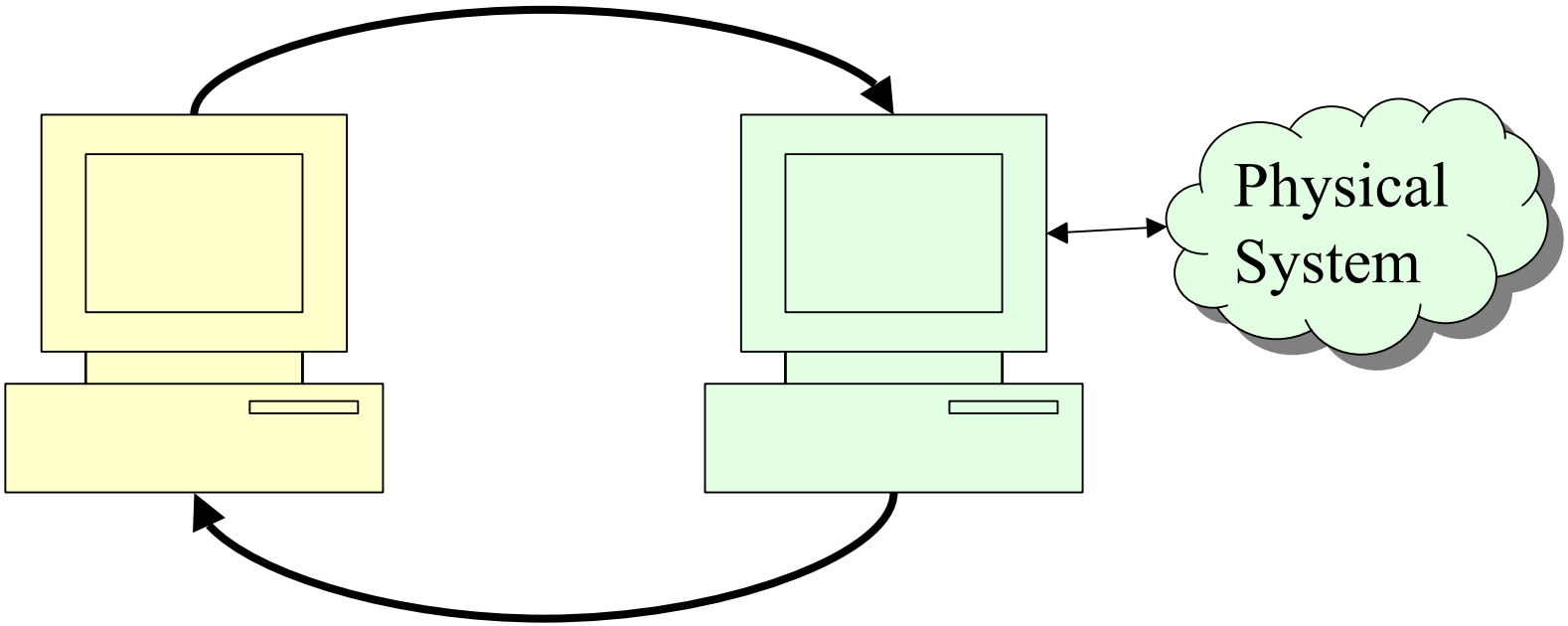
- Process of collecting and generating data is often critical.
  - Current mechanisms for monitoring and control either require physical presence, or use *ad hoc* protocols, formats.
- Instruments and sensors are already “wired”.
  - Usually via non-standard, or perhaps proprietary protocols.
- Using standard mechanisms and protocols can give these devices a grid presence.
  - We then automatically benefit from all the other work on grid standards.
  - Also benefit from a single, unified paradigm, vocabulary, terminology, set of standards for building end-to-end distributed systems.
  - Simplifies end-to-end provenance tracking.
  - Faster, seamless interactions between data acquisition and data processing.
  - Greater interoperability and compatibility.

Philosophy: Push grid standards as close to the instrument or sensor as possible. (But no further!)

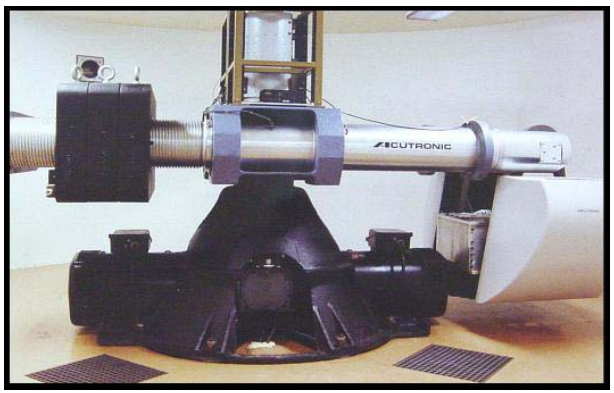
Examples

- Common Instrument Middleware Architecture (CIMA)
  - X-ray crystallography
    - Monitor output remotely to detect bad crystals.
    - Eliminate need to travel to synchrotrons.
    - Assist in provenance.
  - Long Term Ecological Research (LTER) project
    - Automatically update data flows when sensor net changes.
    - Eliminates laborious manual reconfiguration.
    - Assist in provenance tracking.
  - Wireless Sensors (Motes)
    - Battery operated. Limited power, memory, CPU.

Example: NEESgrid Hybrid Experiments



Instruments Controlled Remotely via NEESgrid



NEESgrid Telecontrol Service Requirements

- Uniform interface for heterogeneous systems
  - To enable multi-site experiments
  - Also enables multiple versions of a single experiment:
    - Computation-only (or –mostly) for testing
    - Full scale experiment
    - Small scale experiments for education/outreach
- Impose minimal requirements on control systems.
- Support for recovery from transient errors
- Performance requirements vary widely (1 ms for some experiments; 10 seconds for others).

Issues Raised by NEESgrid

- Instruments must be controlled remotely
- Uniform interface for heterogeneous systems
  - To enable multi-site experiments
  - Also enables multiple versions of a single experiment:
    - Computation-only (or –mostly) for testing
    - Full scale experiment
    - Small scale experiments for education/outreach

NEESgrid Telecontrol Requirements (continued)

- Impose minimal requirements on control systems.
  - Allow sites to continue to use existing systems, some of which are very old
- Security concerns: instruments can cause serious damage or injury
- Support for recovery from transient errors
- Performance requirements vary widely (1 ms for some experiments; 10 seconds for others).

NEESgrid Teleoperation Control Protocol (NTCP)

- Transaction-based
- State model
  - provides for at-most-once execution
  - guarantees that a control point is doing at most one thing at a time
  - guarantees that requests involving any control point are executed in the order received
- Protocol allows for negotiation of request parameters prior to execution.

Example: NEON

- National Ecological Observatory Network
- Will be a national-scale facility for the study of:
  - Biodiversity
  - Biogeochemical Cycles
  - Climate Change
  - Hydroecology
  - Infectious Disease
  - Invasive Species
  - Land Use
  - Emerging Issues

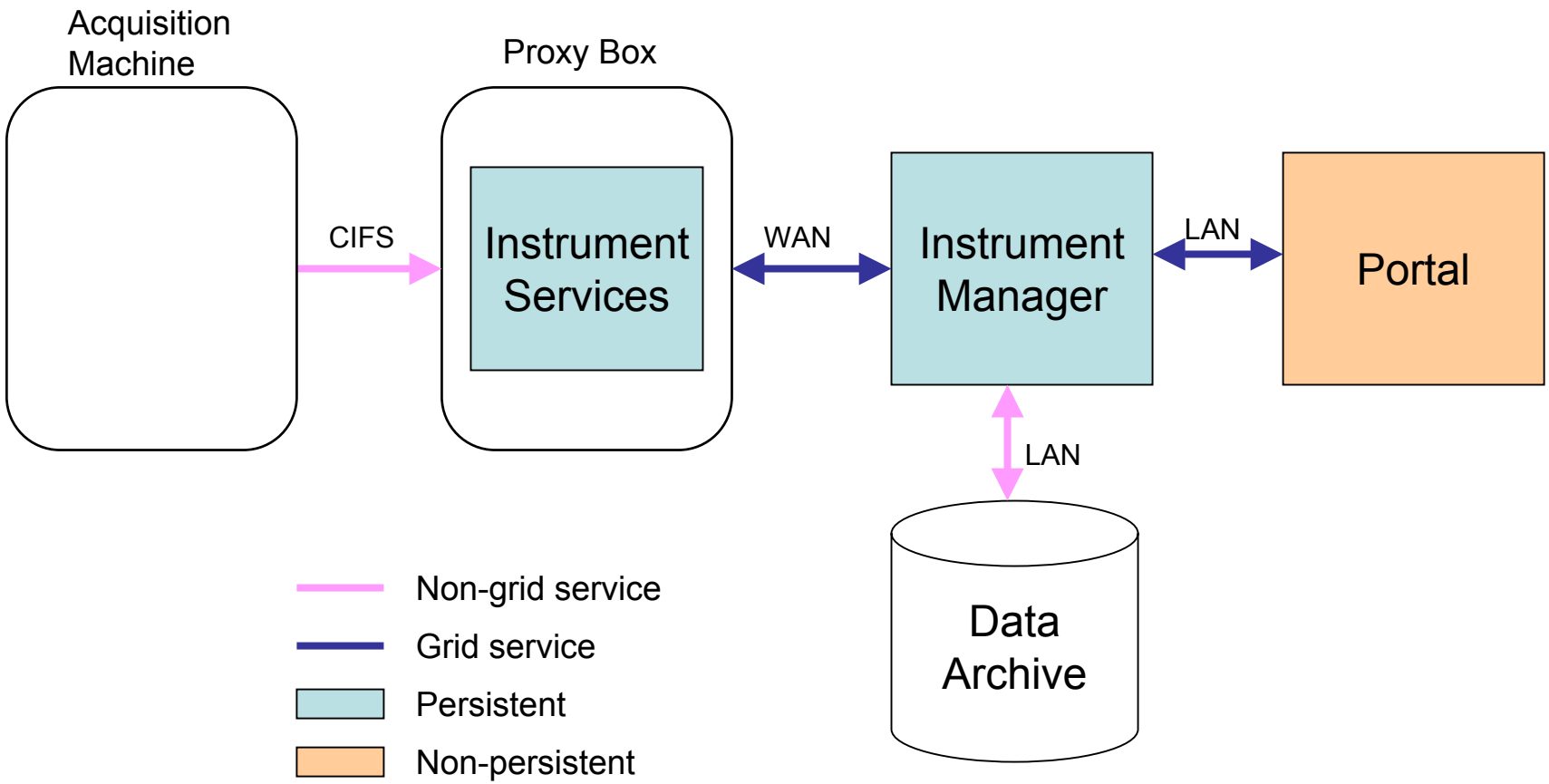
Issues Raised by NEON

- Very diverse application space
- Many kinds of sensors involved
- Sensor data must be understood by scientists in several different fields
- Sensors may be placed in remote locations
  - Must be remotely configurable
  - Must be able to deal with unreliable or low-bandwidth networks
- Some sensor data is confidential

Common Instrument Middleware Architecture (CIMA)

- Develop a set of standard grid services for accessing and controlling sensors.
  - Based on Web standards such as WSDL, SOAP, XML, etc.
- Develop a sensor ontology for describing sensors.
  - Applications use the description to interact.
- The goal is to develop middleware that abstracts and layers functionality.
  - Minor differences in sensors should only result in minor loss of functionality to the application.

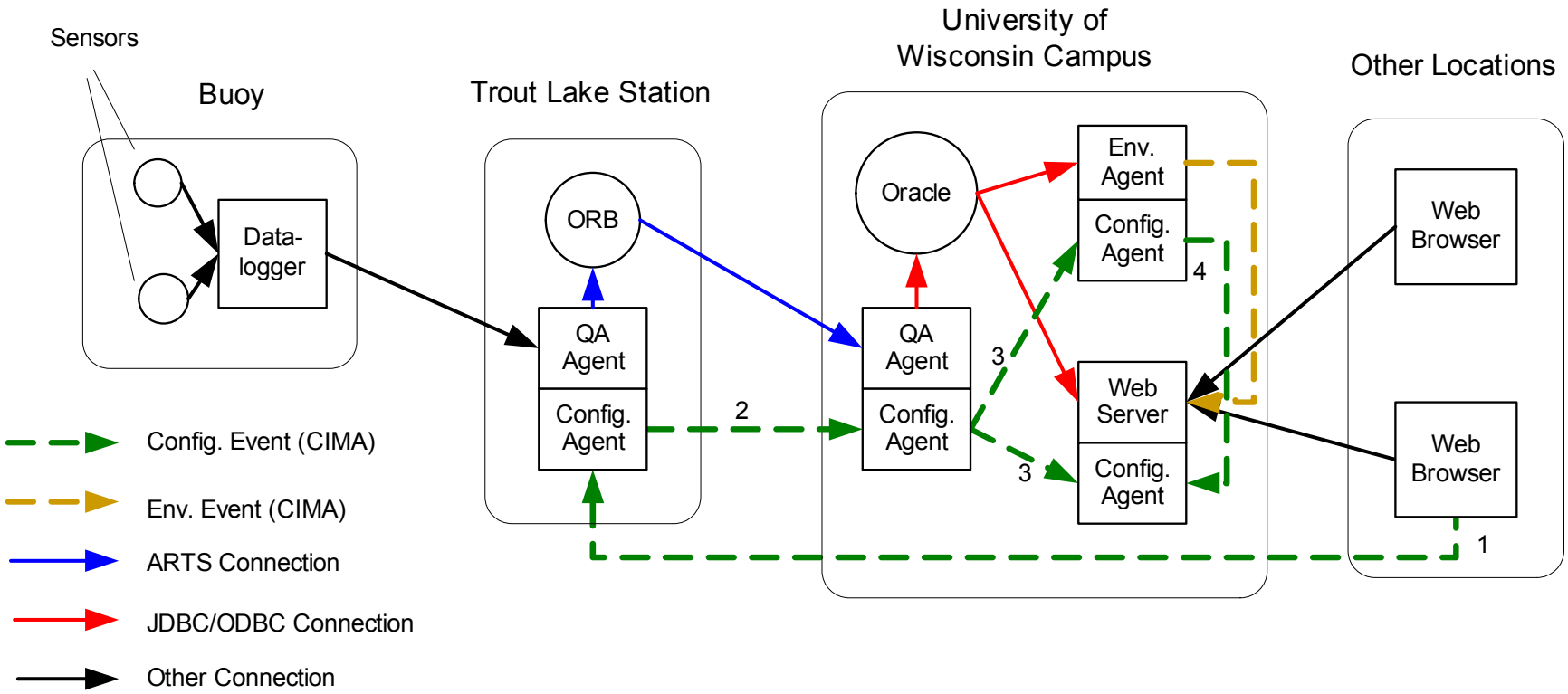
X-Ray Crystallography



X-Ray Crystallography

- Scientists are very reluctant (understandably) to install your software on the acquisition machine.
  - Use a proxy box by which to access files via CIFS or NFS.
  - Scan for files which indicate activity.
    - Unfortunately, scientists can manually create files, which can confuse the scanner. No ideal solution.
- For sensor data, request-response is not ideal.
  - Push data using one-way messages.
    - In WSDL 2.0, consider “connecting” out-only services to in-only services.

Long-Term Ecological Research (LTER)



LTER

Grid service standards are *extensible*. Take advantage of this fact.

- Networks and machines may be unreliable. Also, much expertise in pre-existing systems such as Antelope (ARTS).
  - Antelope is mainly about transport. SOAP is designed to be bound to transport other than HTTP. Use Antelope to transport SOAP messages.
- Sensors may come in different makes and models.
  - Use an ontology to describe the sensor.

Wireless Sensors (Motes)

- Performance
  - Conformance to standards does not require the use of heavyweight toolkits with large memory footprints.
    - SOAP can be generated with a simple `sprintf()`.
    - SOAP can be parsed in an *ad hoc*, application-specific manner.
      - Creates a tighter coupling, but better than using a custom format.
  - Binary XML can be used to even further reduce the communication requirements.

Summary

- Grid services and web services are designed to be extensible.
  - Transport protocols can be changed.
  - Wire formats can be changed.
- Grid services do not require heavyweight toolkits.
- No magic bullet for integration into existing infrastructure.
  - But a solution of some kind usually exists.