# NEESGrid - A Grid Portal Study

Charles Severance

University of Michigan

NEESGrid SI Team

www.neesgrid.org

# Acknowledgements

This presentation is based on materials from many members of the NEESGrid System Integration team including: Bill Spencer, Carl Kessleman, Tom Finholt, Lee Liming, Laura Perlman, Paul Hubbard, Joe Futrelle, Kincho Law, Jun Peng, Greg Fenves, Tomasz Haupt, Jim Myers, Doru Marcusiu, Randy Butler, Jim Eng and many others.

# NEES Founding

- George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES).
- Funded in 1999 - > $100M
- Goal: Transform the nation's ability to carry out earthquake engineering research, to obtain information vital to develop improved methods for reducing the nation's vulnerability to catastrophic earthquakes, and to educate new generations of engineers, scientists and other specialists committed to improving seismic safety.
- To be Completed: October 2004

- NEESgrid facilitates research capabilities previously unavailable
- NEESgrid links earthquake researchers across the U.S. with leading-edge computing resources and research equipment and allowing collaborative teams (including remote participants) to plan, perform, and publish their experiments
- NEESgrid is a coordinated and secure architecture/environment
- NEESgrid is a modular and extensible environment with a customizable user interface
- NEESgrid provides common tools that allow leveraging resources and experiences
- Rather than having to worry about the required cyber infrastructure, NEESgrid allows researchers to focus on the earthquake engineering challenges at hand
- The goal of the System Integrator (SI) is to develop NEESgrid as the Cyber Infrastructure that will facilitate this next generation of experimentation/simulation in earthquake engineering

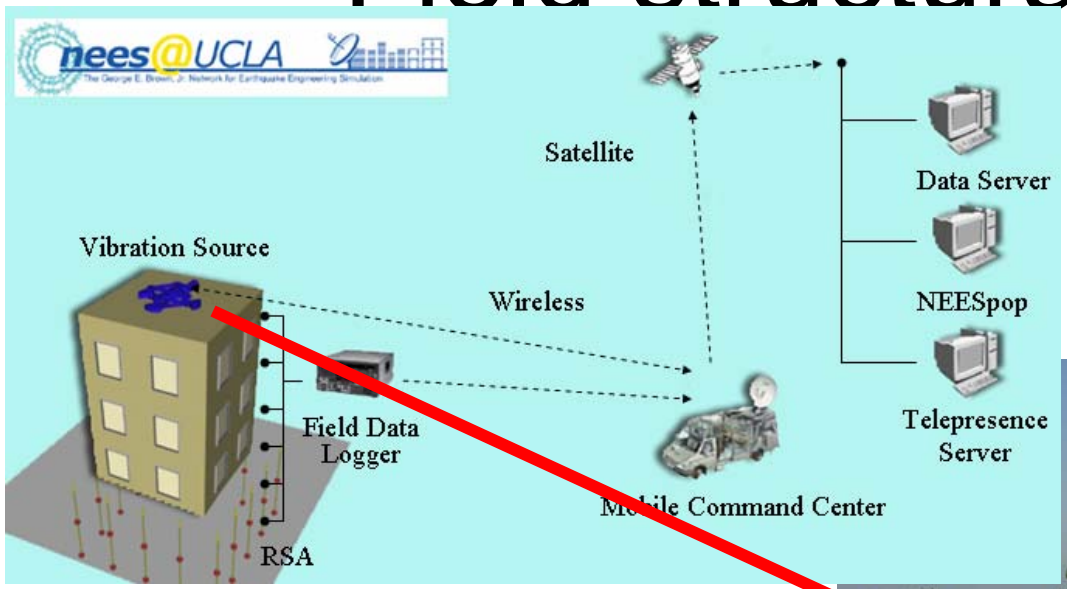
# NEES Components

- New experimental facilities (15)
  - Oregon State University, Rensselaer Polytechnic Institute, University of Buffalo, University of Colorado at Boulder, University of Minnesota, University of Nevada at Reno, University of Texas at Austin, and the University of California campuses at Berkeley, Davis and Los Angeles
- Collaborative Software System: NEESGrid
  - Collaboration
  - Data capture and sharing
  - Tele-presense and Tele-operation
  - Simulation
  - Support for Hybrid Simulation and Physical Experiments

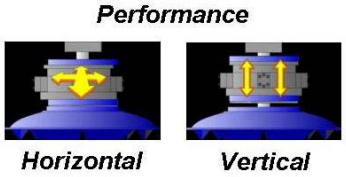
# Centrifuge: UC Davis

# Wave basin: Oregon State

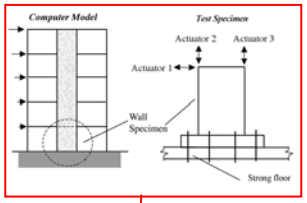
# Field structural: UCLA

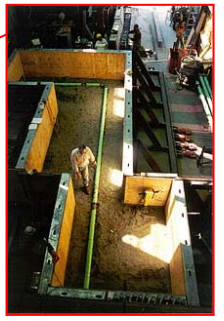


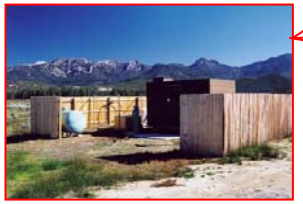

# Field geotechnical: Texas

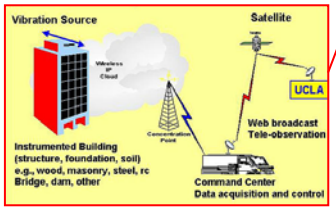



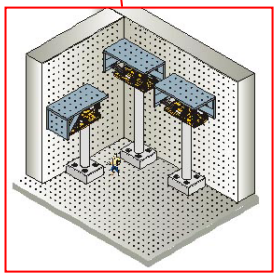

Multiple Shake Table Research Facility
University of Nevada, Reno
Large-Scale Structures Laboratory


Computer Model
Test Specimen
Actuator 2
Actuator 3
Actuator 1
Wall Specimen
Strong floor






Vibration Source
Satellite
Wireless IP Cloud
UCLA
Concentration Point
Web broadcast
Tele-observation
Instrumented Building
(structure, foundation, soil)
e.g., wood, masonry, steel, rc
Bridge, dam, other
Command Center
Data acquisition and control



# If we build it, they will collaborate

- Data and access to data represent fundamental barriers to dispersed collaboration
- Efficient movement of vast amounts of data is a prime rationale for cyberinfrastructure
- Federating, visualizing and mining data are principle challenges

# NEES Facilities

# NEESGrid Software

- Founding NMI Technologiess
  - Globus Toolkit
  - OGCE Collaboration Toolkit
- New Work
  - Data and Metadata Repository - NCSA
  - Data Acquisition, Storage, and Visualization
  - Simulation Portal

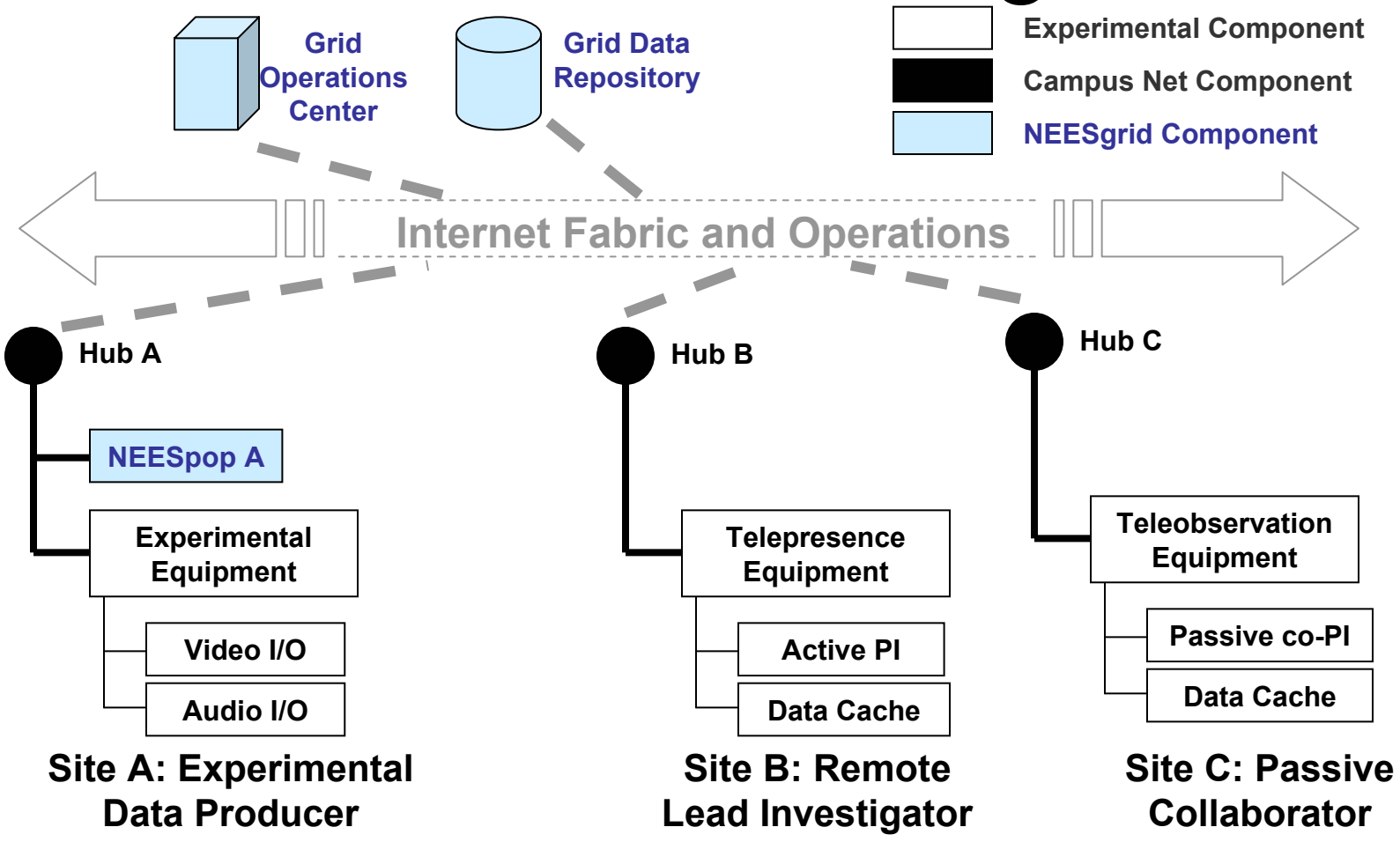
# NEESGrid Partners

- Argonne National Labs
  - Globus toolkit, Data Acquisition, Telepresense
- Information Sciences Institute (USC)
  - Globus toolkit, Teleoperation and Telecontrol
- National Center for Supercomputing Applications (NCSA)
  - System Integration, Data Repository
- University of Michigan
  - Collaborative Grid Portal, Data Modelling, Visualization, Video as Data

# NEESGrid Partners

- Stanford
  - Data Model Design
- Mississippi State University
  - Simulation Portal
- University of California Berkeley
  - OpenSEES and FedeasLab
- Pacific Northwest National Laboratory
  - Scientific Annotation Middleware (SAM), Electronic Notebook

# The *Grid* in NEESgrid



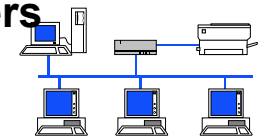

**Site A: Experimental Data Producer**

**Site B: Remote Lead Investigator**

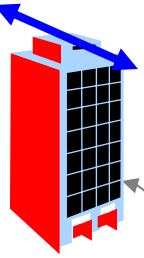
**Site C: Passive Collaborator**

# NEES Resources

Remote Users
(Faculty, Students, Practitioners)

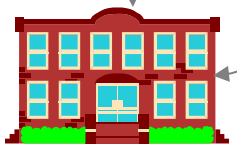


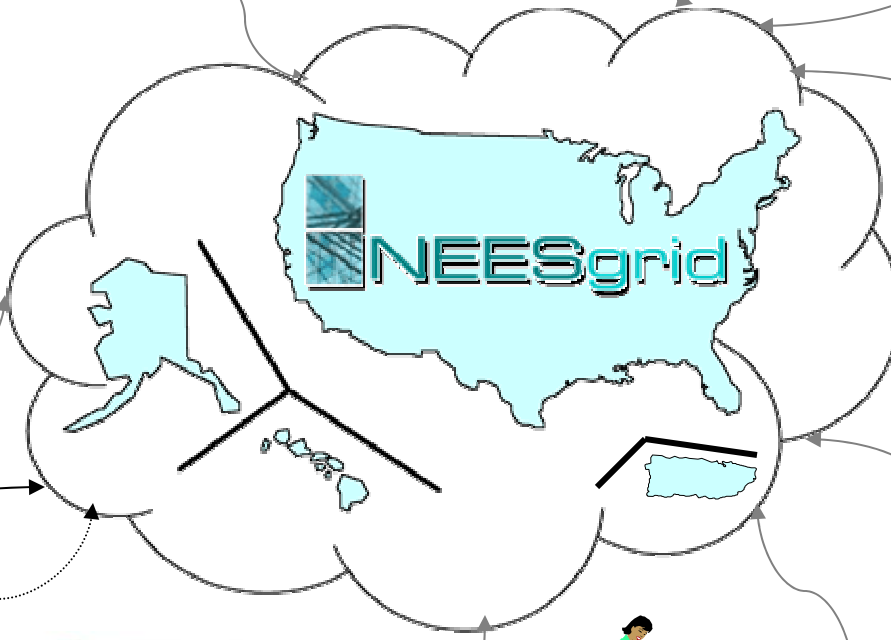
Instrumented Structures and Sites

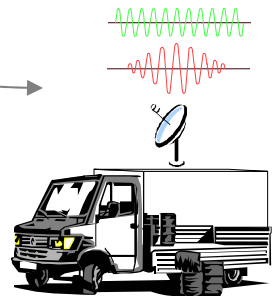


Simulation Tools Repository

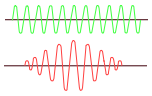
Laboratory Equipment



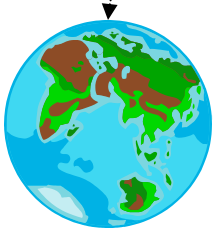


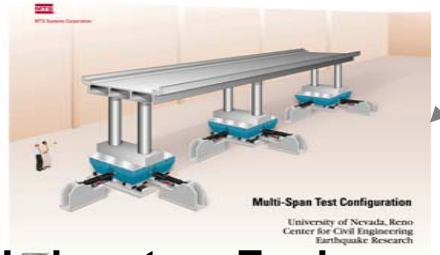
Field Equipment



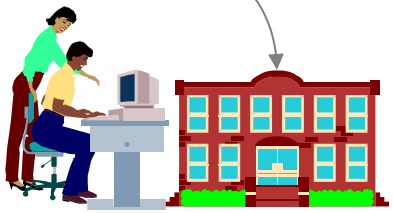
Curated Data Repository



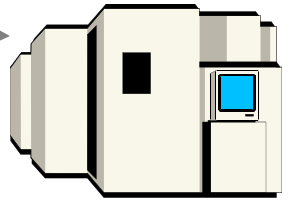
Global Connections




Laboratory Equipment



Remote Users:
(K-12 Faculty and Students)



Leading Edge Computation

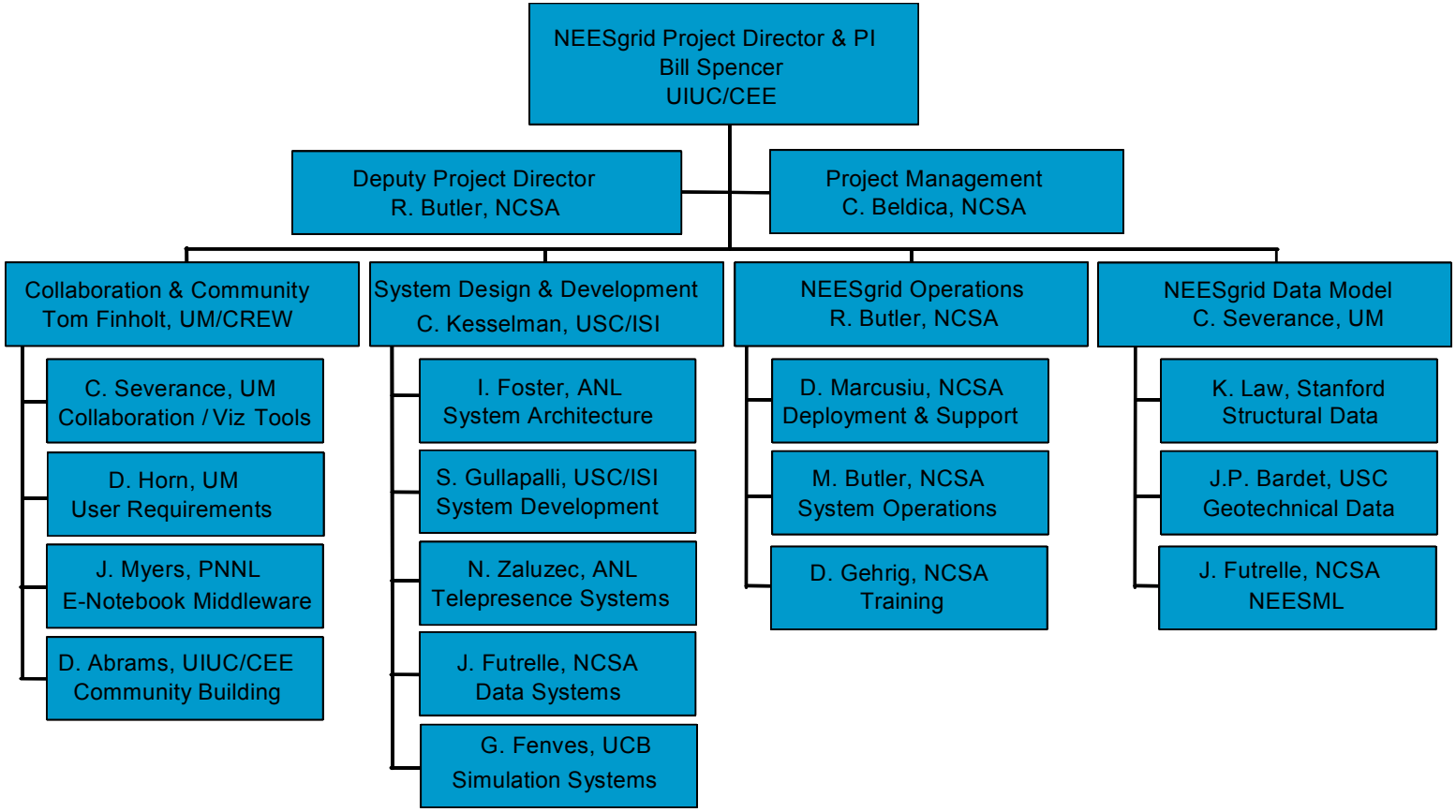
# The Main Components of NEESgrid

- Tele-Control Services
- Tele-Observation and Data Visualization
- E-Notebook
- Streaming Data services
- DAQ and related services
- Data and Metadata services
- Remote Collaboration and Visualization tools and services
- Core Grid Services, deployment efforts, packaging
- Computational Simulation component

# The NEES Win

- *New engineering capabilities*
  - *rapid assembly of virtual teams*
  - *access to remote facilities and experiments*
  - *interfaces to distributed data archives/experiment repositories*
- National and international cyberinfrastructure leverage
  - corporate and government commitments
  - billions of dollars in leverage
    - commoditization of infrastructure
- Distributed facility and collaboration access
  - NEES equipment sites (ES) and distributed collaborators
  - cooperating institutions and policies
- Strong security features
  - secure experiment control and data sharing policies
- Resource discovery and monitoring services
  - available resource identification and continuous status monitoring

# Organizational Chart (January 2004)



# NEES Architecture

# The Role of the NEESgrid System Architecture

- Define the core capabilities of NEESgrid
- Facilitate interoperability, extensibility and scalability
- Provide a foundation on which the diverse NEES usage scenarios can be supported
  - Not single point solution

# Architecture Approach

- Common infrastructure that can used across all NEES applications
  - Balance generic mechanisms, extensibility for future growth, efficiency for application specific tasks
- Validate against user requirements
  - Input from user requirements analysis
  - MOST, EBD build on proven technology base

# NEESgrid and the Grid

- Grid is infrastructure to support
  - Data sharing, numeric simulation, remote observation and control, collaboration
- Maps well into NEES requirements
  - Similarity of problem space and objectives
- Synergistic with many other projects
  - E.G. SCEC, ETF, …
  - Minimizes risk

# Open Grid Services Architecture

- Builds on Web Services technology
  - A Grid service is a Web service with extras
- Significant industry buy in
  - IBM, HP, Oracle, SGI, …
- High-quality open source implementation
  - Globus Toolkit®

# NEESGrid and NSF Middleware Initiative

- CISE program to harden, test and support national middleware infrastructure
- Significant NMI presence in Grid space
- Plan to eventually fold NEES specific services into NMI releases

# Software Components

- Extant software
  - particularly significant elements of the NSF Middleware Initiative (NMI) software system
- Custom software to address general NEESgrid issues
  - Produced by SI team
- Site-specific, and application specific software
  - to be produced by the equipment sites, other NEES participants, or other sources.

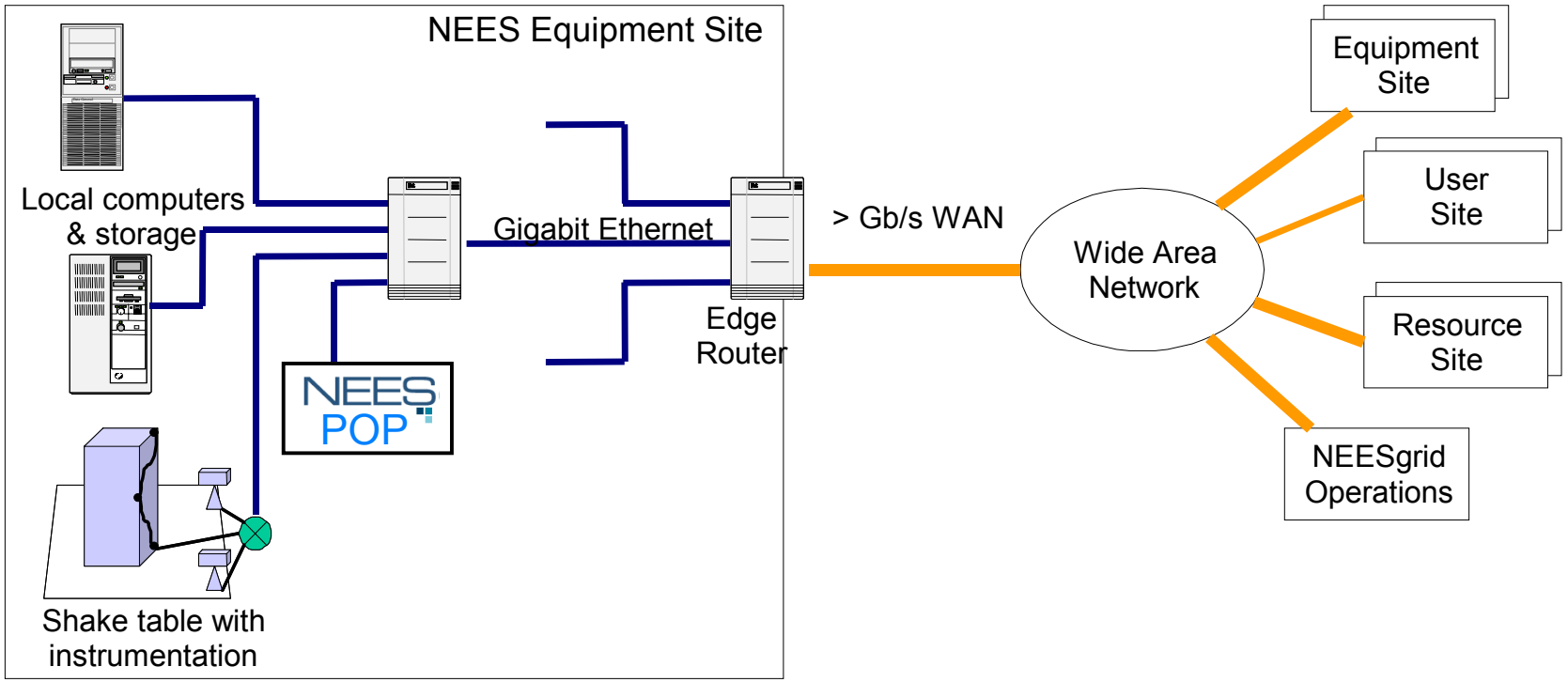
# Physical Elements

- A moderate number of *equipment sites*,
- A moderate number of *resource sites*,
  - data repositories and/or computer systems
- A potentially large number of *users*
  - including earthquake engineers, students, and others.
- Campus and wide area *networks*
- An *operations center*,
  - provides monitoring and diagnostic facilities for NEESgrid as a whole

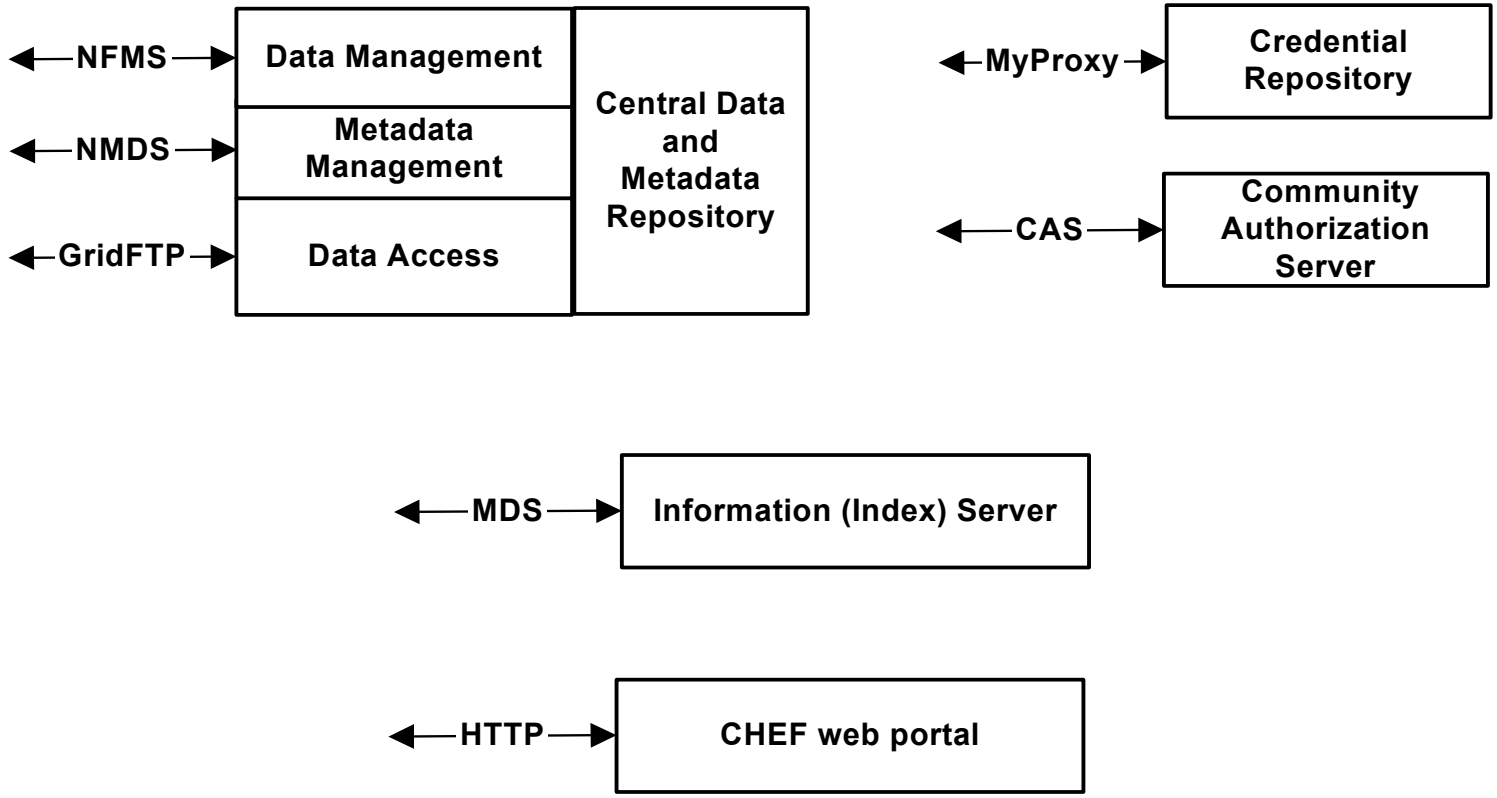
# NEESgrid Core Capabilities

- Tele-control and tele-observation of experiments
- Data cataloging and sharing
- Remote Collaboration and visualization tools and services
- Simulation execution and integration

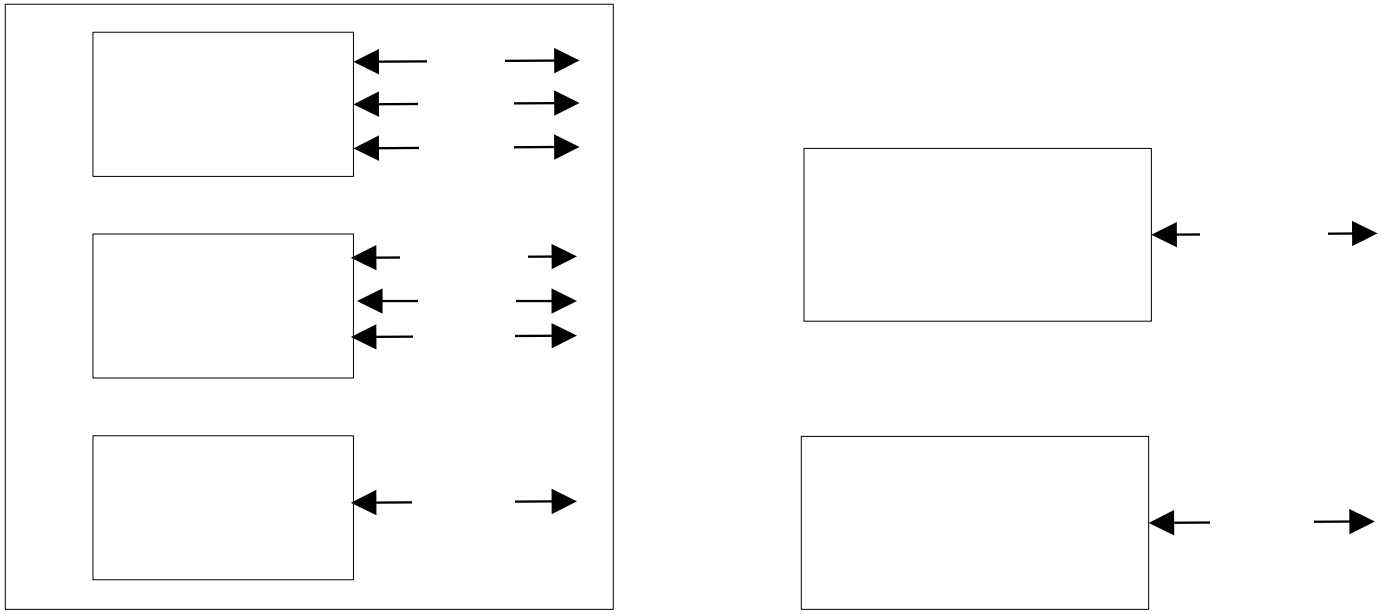
# NEESgrid High-level Structure



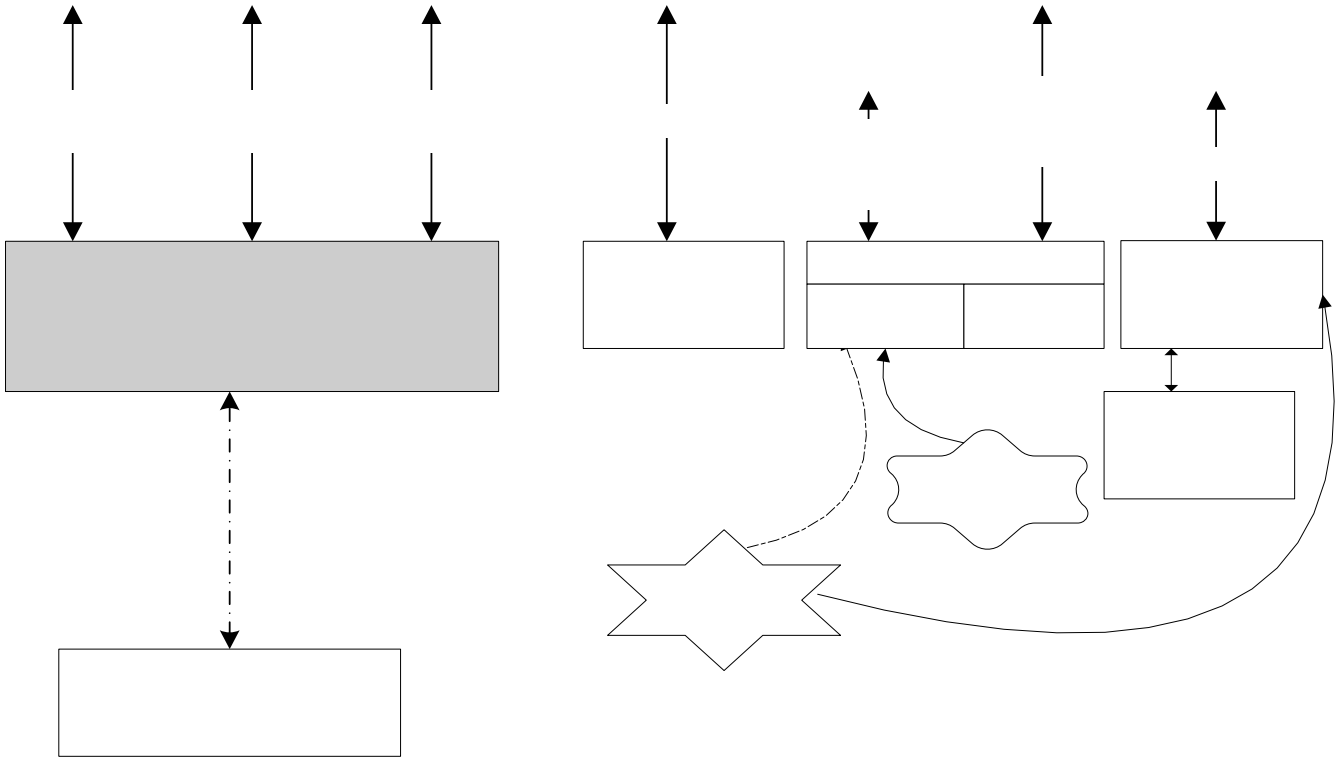
# Centralized NEES-Wide Services



# Non-Centralized NEESgrid Services



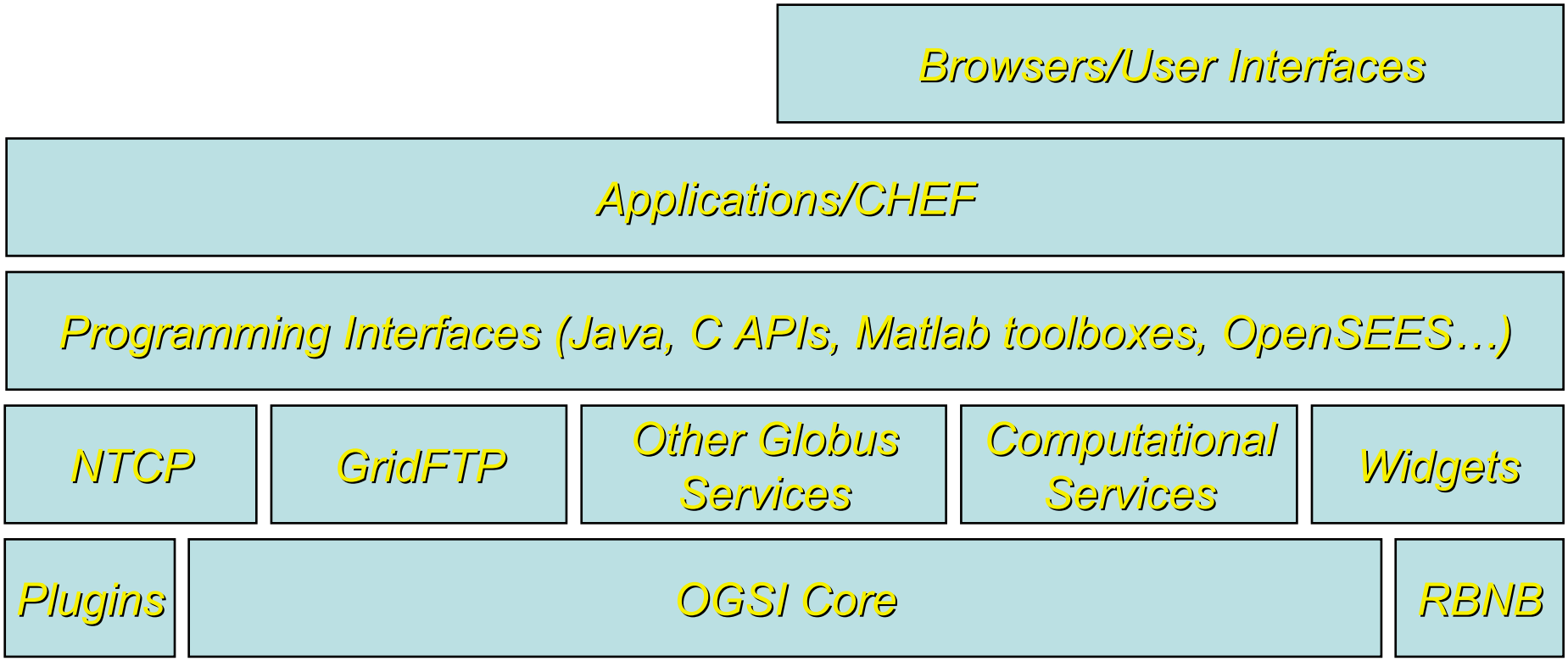
# Architecture of NEESgrid Equipment site.



# Globus Toolkit V3

- High quality open source OGSI implementation
  - Developed by The Globus Alliance
- Commercial support available
- Globus services include:
  - Security
    - Authentication and authorization
  - Status and configuration
  - Resource management
  - Data services
    - Data movement
    - Data access

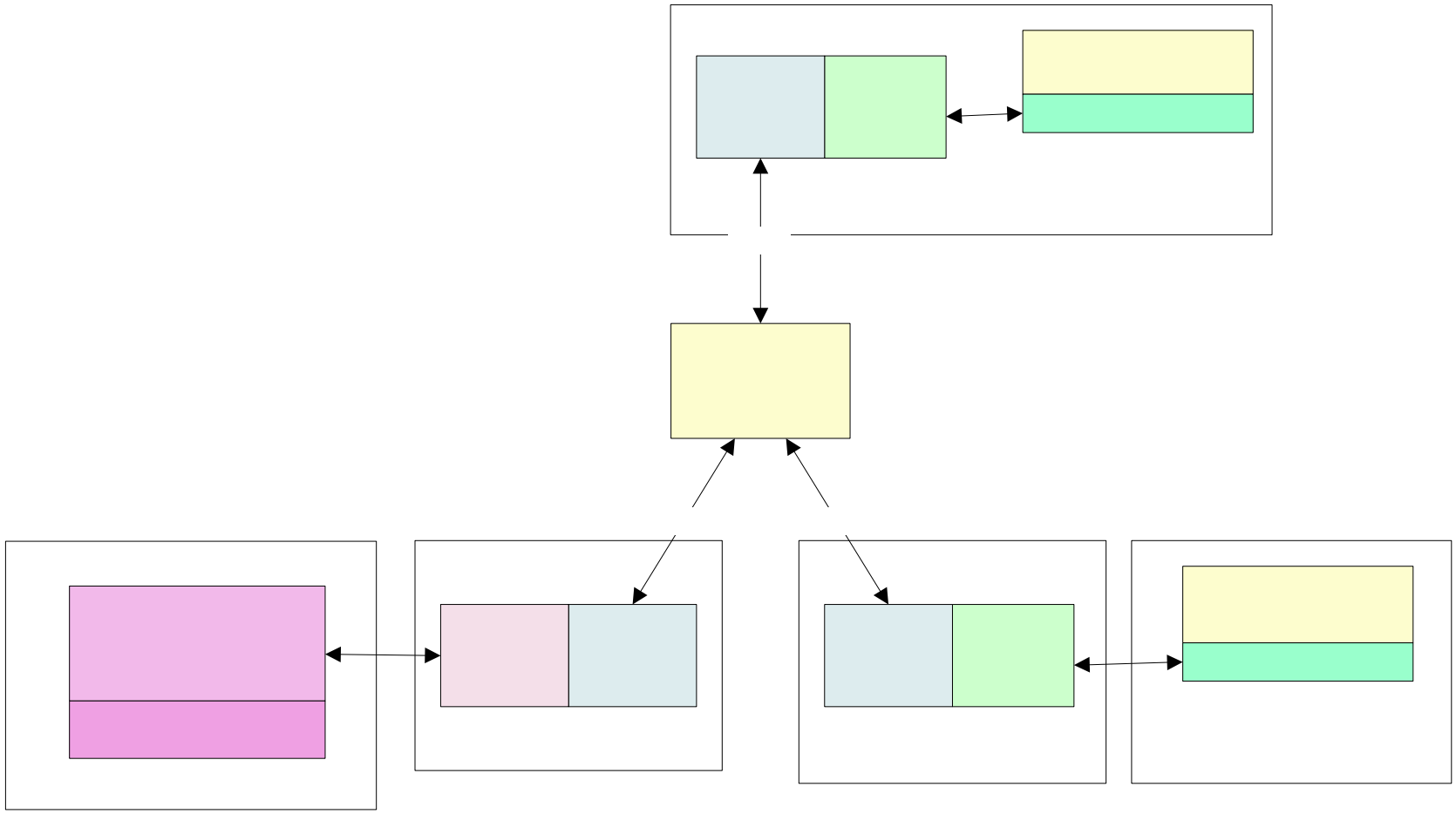
# NEESgrid Software Stack



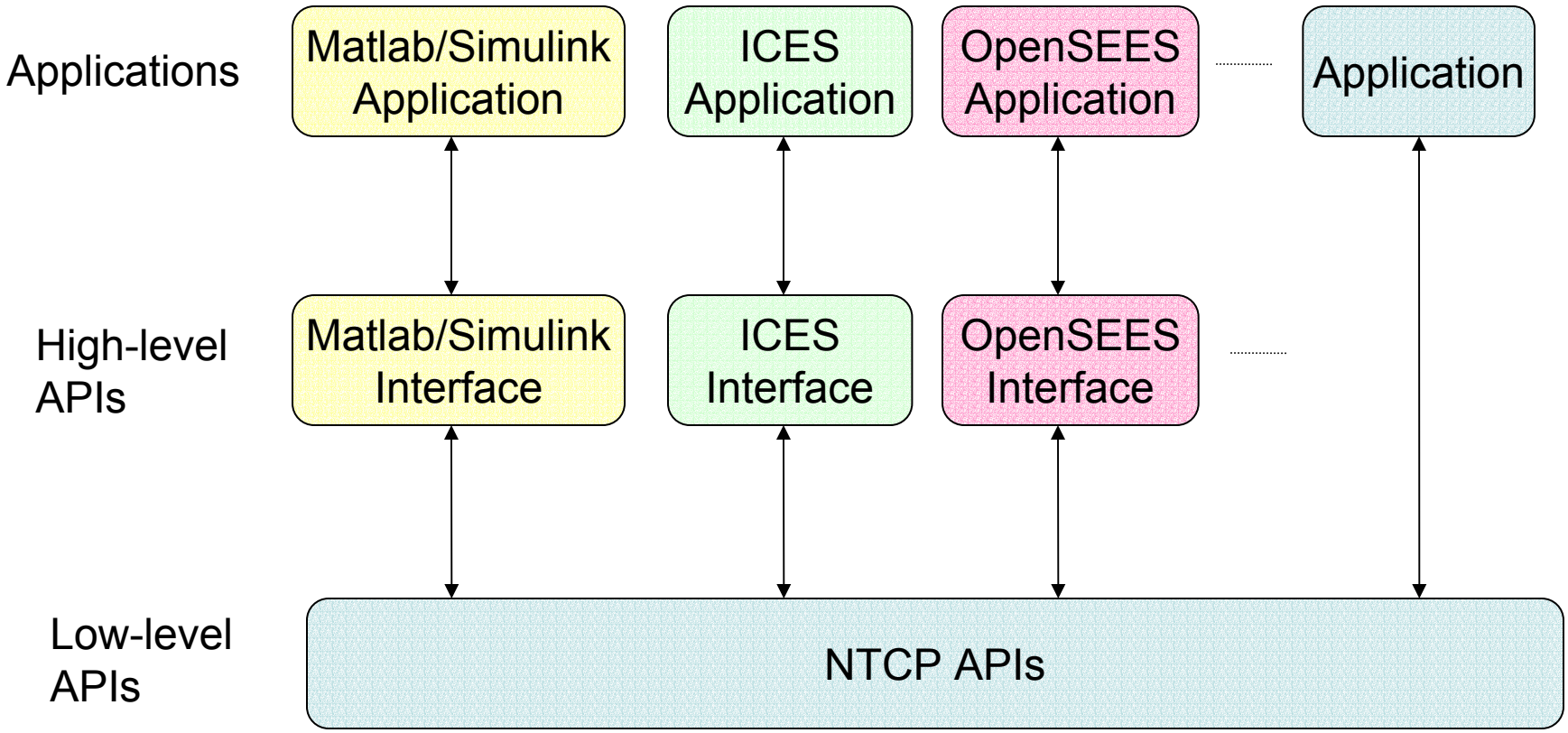
# Tele-Control Services

- A single, transaction-based protocol and service (NTCP) to control physical experiments and computational simulations.
- OGSI based implementation (GT3.0)
- Plug-ins to interface the NTCP service
  - A computational simulation written in Matlab
  - Shore Western control hardware
  - MTS control hardware (via Matlab and xPC)
  - Labview
  - C
- Security architecture, including GSI authentication and a flexible, plug-in-based authorization model.

# Plug-in approach



# Programming Interfaces

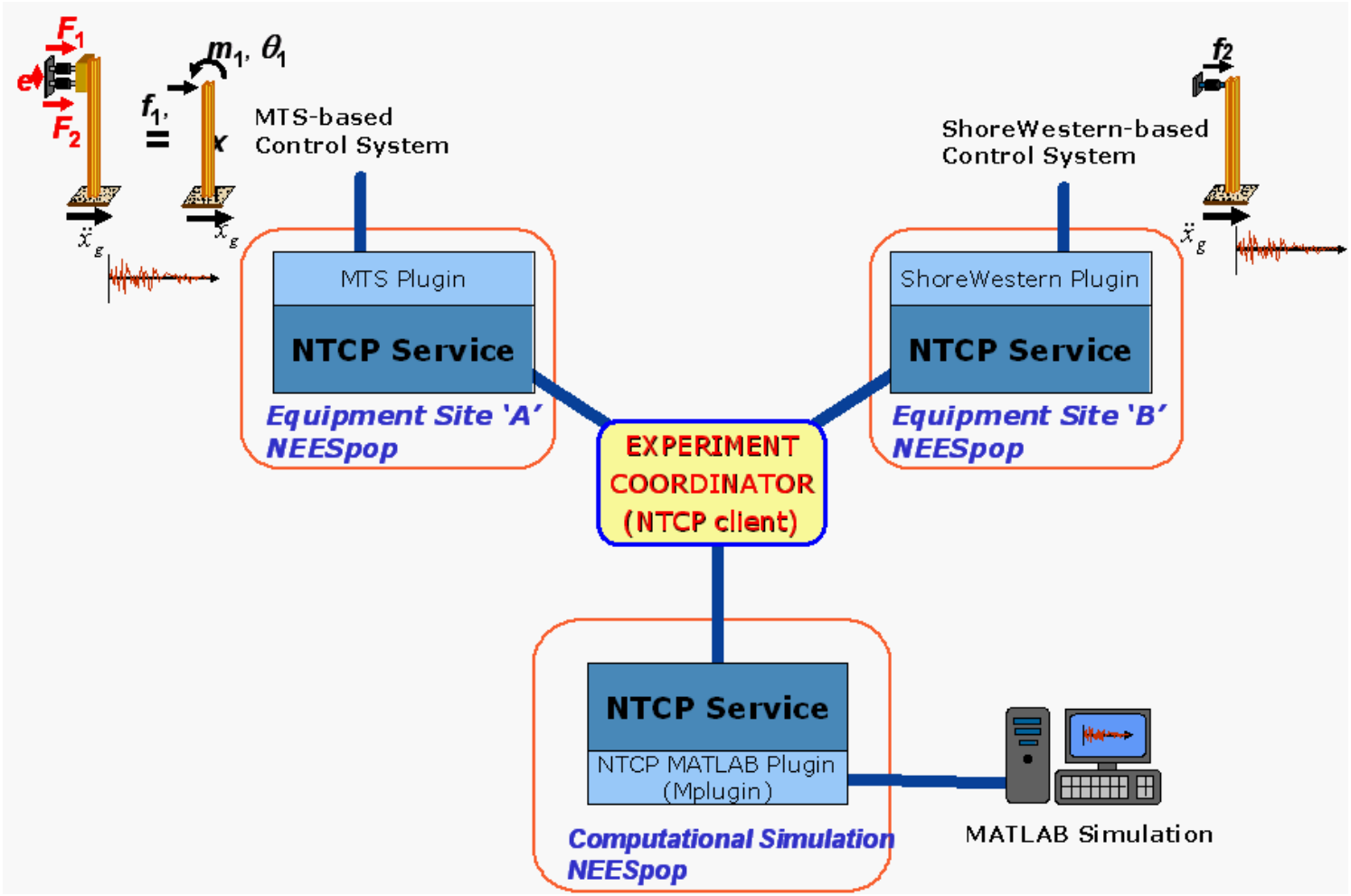


# NEES TeleControl Protocol (NTCP)

# NEESgrid Core Control Components

- A uniform control interface for both physical and simulation components is achieved through a single control architecture.
  - NTCP Service
  - NTCP Client APIs
  - NTCP Plugin APIs
- Overall, control components are well-defined and available. Equipment sites are installing and configuring their control capabilities through our EBD program.

# NTCP Service in Context



# High-level NTCP Service Features

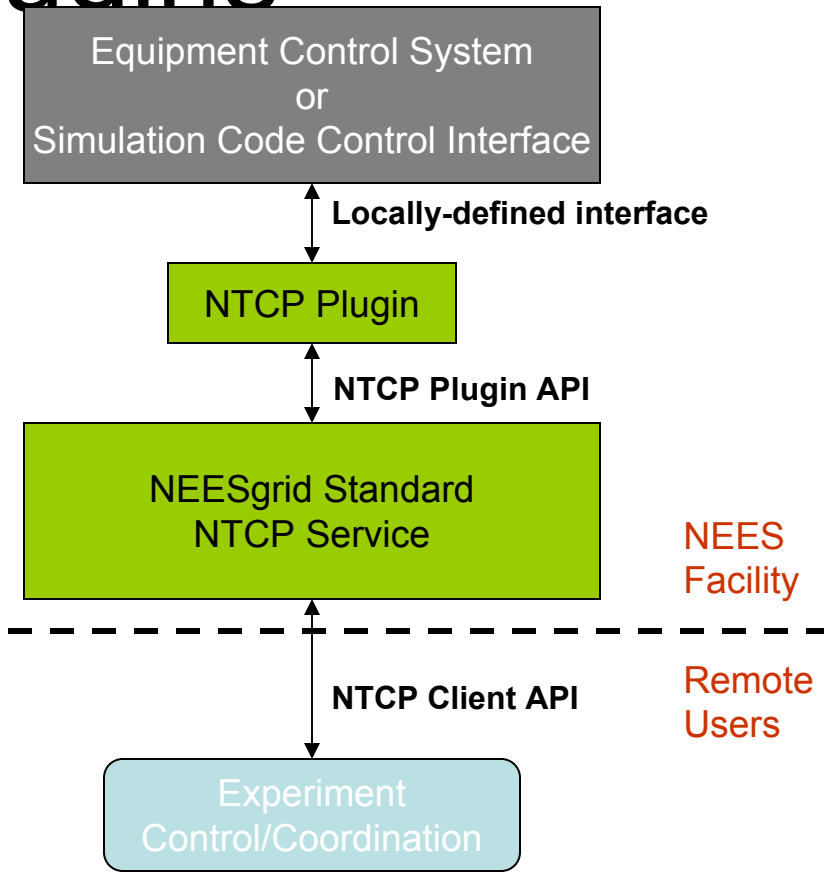
- Two-stage control system (propose, execute)
  - Satisfies key equipment site requirements (safety, protection of equipment investments)
- Reliability & robustness features
  - Allow client and server to recover from unusual/failure states
- Plugin architecture
  - Isolates site-specific code from NEESgrid-standard NTCP service code
- OGSI-compliance
  - Ensures that NEESgrid interoperates with other Cyberinfrastructure components (through compatible security and service frameworks)

# NTCP Client APIs

- NTCP Client APIs allow software to control a physical or simulation component via the NTCP service.
- NTCP Client APIs are available, are documented, and are in use.
  - Java Client and Java “Helper” APIs are available and were used by Chef in MOST and MiniMOST. These are also used in NEESgrid acceptance testing and will be used in upcoming EBD activities.
  - C/C++ Client API is available for early adopter use. This will be used in upcoming EBD activities.

# NTCP Plugins

- An NTCP Plugin links the NTCP Service to the local control system or simulation component.
  - The NTCP Plugin API (available in Java and C/C++) is documented and example Plugins are available for use.
  - Ultimately, it's the equipment site's or simulation code developer's responsibility to "hook up" their components to the NEESgrid core control service.
  - The SI team has developed and tested a number of NTCP Plugins, resulting in many options and examples.
  - Some equipment sites have begun developing their own NTCP Plugins.
- NTCP Plugins have been used in a number of settings.
  - MOST Experiment
  - MiniMOST
  - EBD activities
  - Acceptance Testing and Equipment Site Validation



# NTCP Plugins Developed by SI

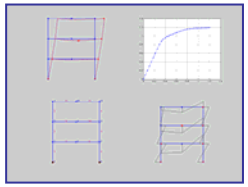
- **Dummy Plugin**
  - **Unit testing, Equipment Site Validation**
- **Mplugin + Matlab NTCP Toolbox**
  - **Matlab control systems and simulation components (e.g., MOST experiment)**
- **LabView Plugin**
  - **LabView control systems, MiniMOST, Still digital camera control**
- **C Gateway Plugin + C Plugin API**
  - **Supports Plugins written in C/C++**
- **ShoreWestern Plugin**
  - **UIUC components in MOST experiment**

# NEESGrid Simulation

# NEESgrid Simulation Team

- G.L. Fenves, UC Berkeley
- F. McKenna, UC Berkeley

- F.C. Filippou, UC Berkeley



- T. Haupt
  Mississippi State Univ.

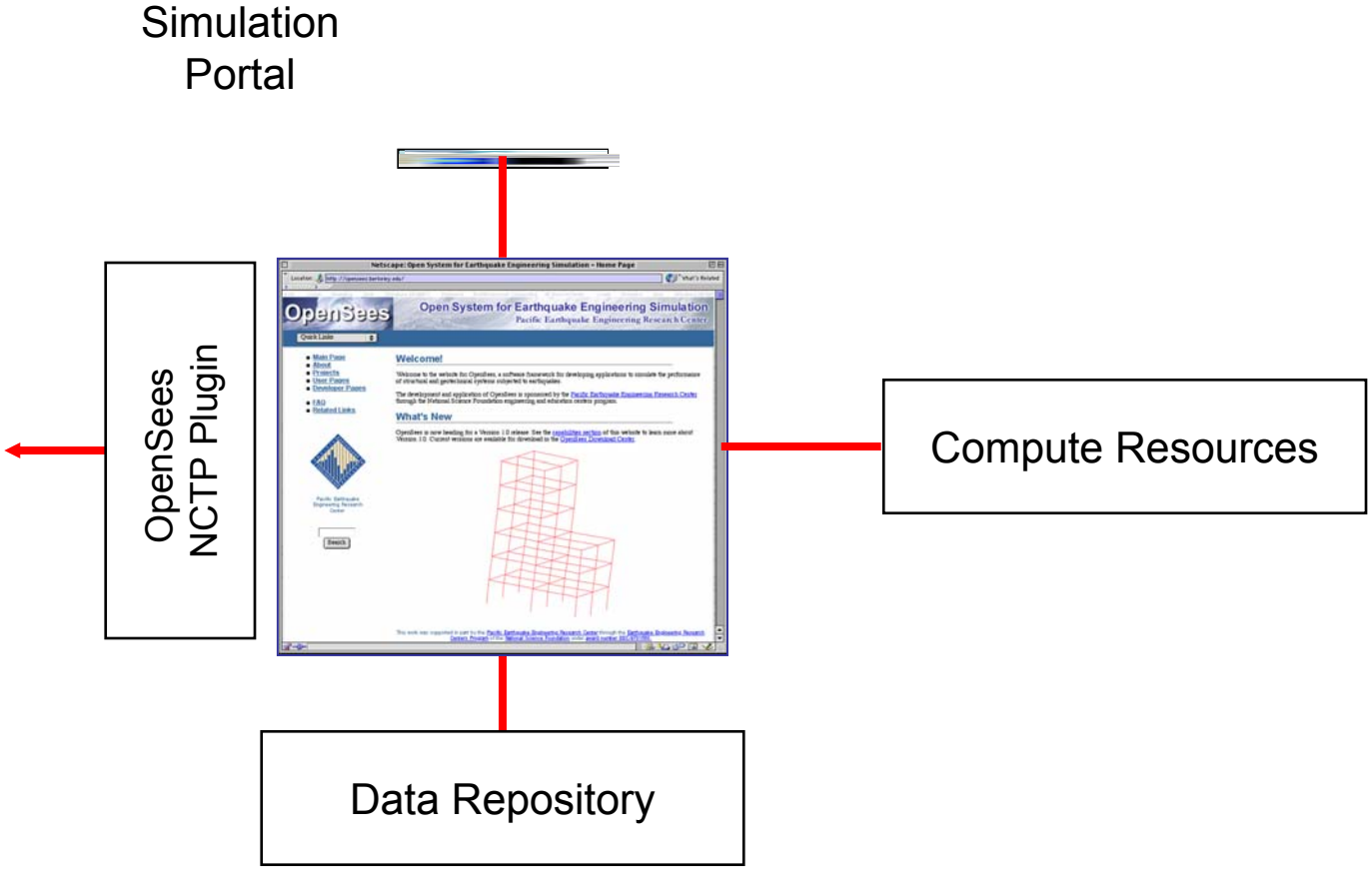
- B. Spencer

NEESgrid

# Simulation Component Objectives

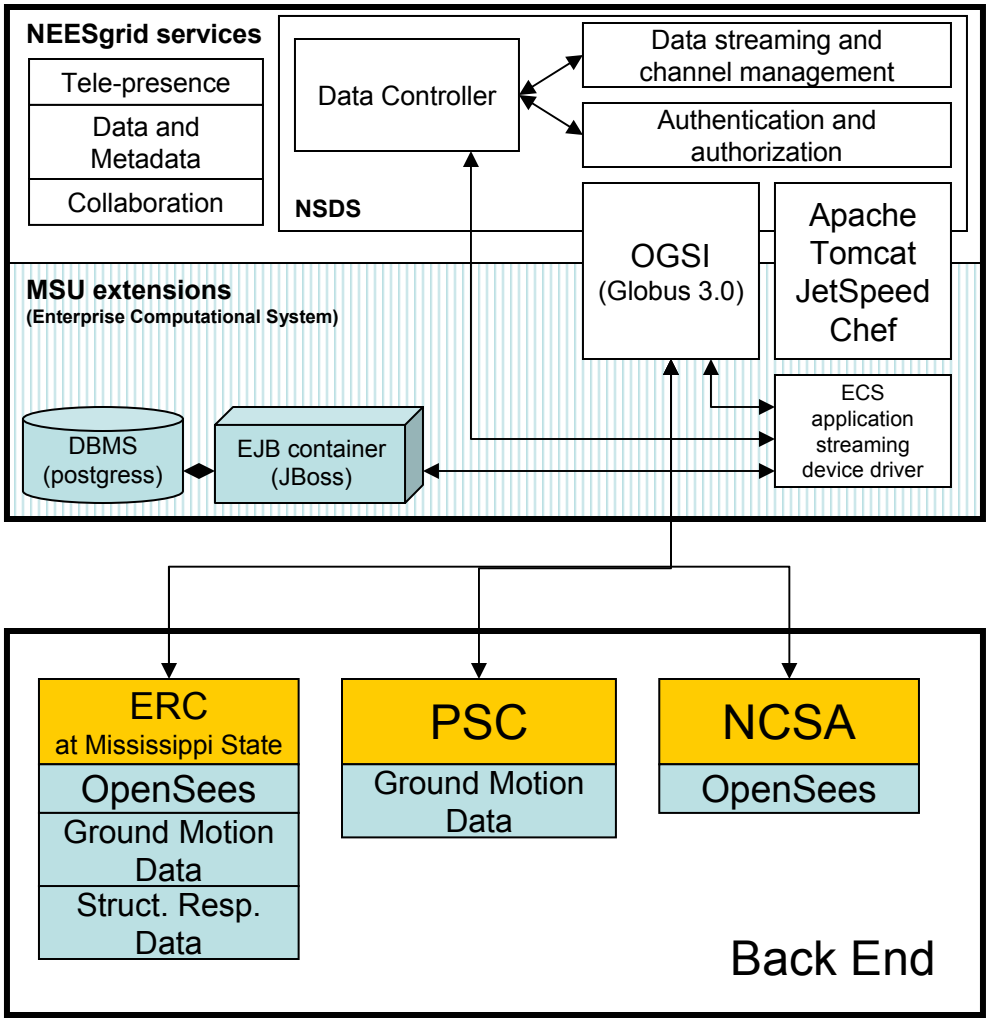
- Provide capability for modeling and simulation of structural and geotechnical systems within NEESgrid.
- Create NEES open-source *community* for simulation software for future simulation application development.
- Provide interfaces from simulation software to NEESgrid data repositories using appropriate data models.
- Provide portal access to NEESgrid or other high-end compute resources.
- Provide Matlab framework for research, prototyping, and education in simulation.

# NEESgrid Simulation Overview



# NEESport Architecture

NEESpop (middleware)

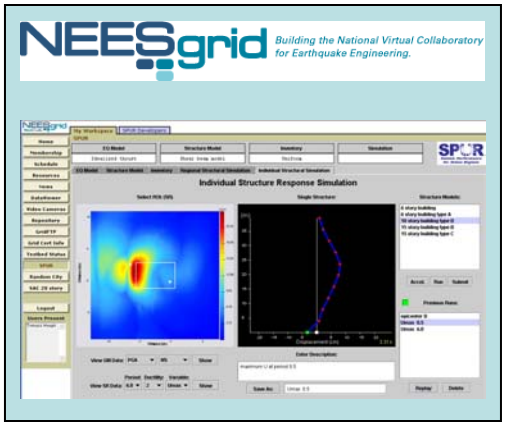



Data

SPUR applet

Request

Front End




NEESgrid

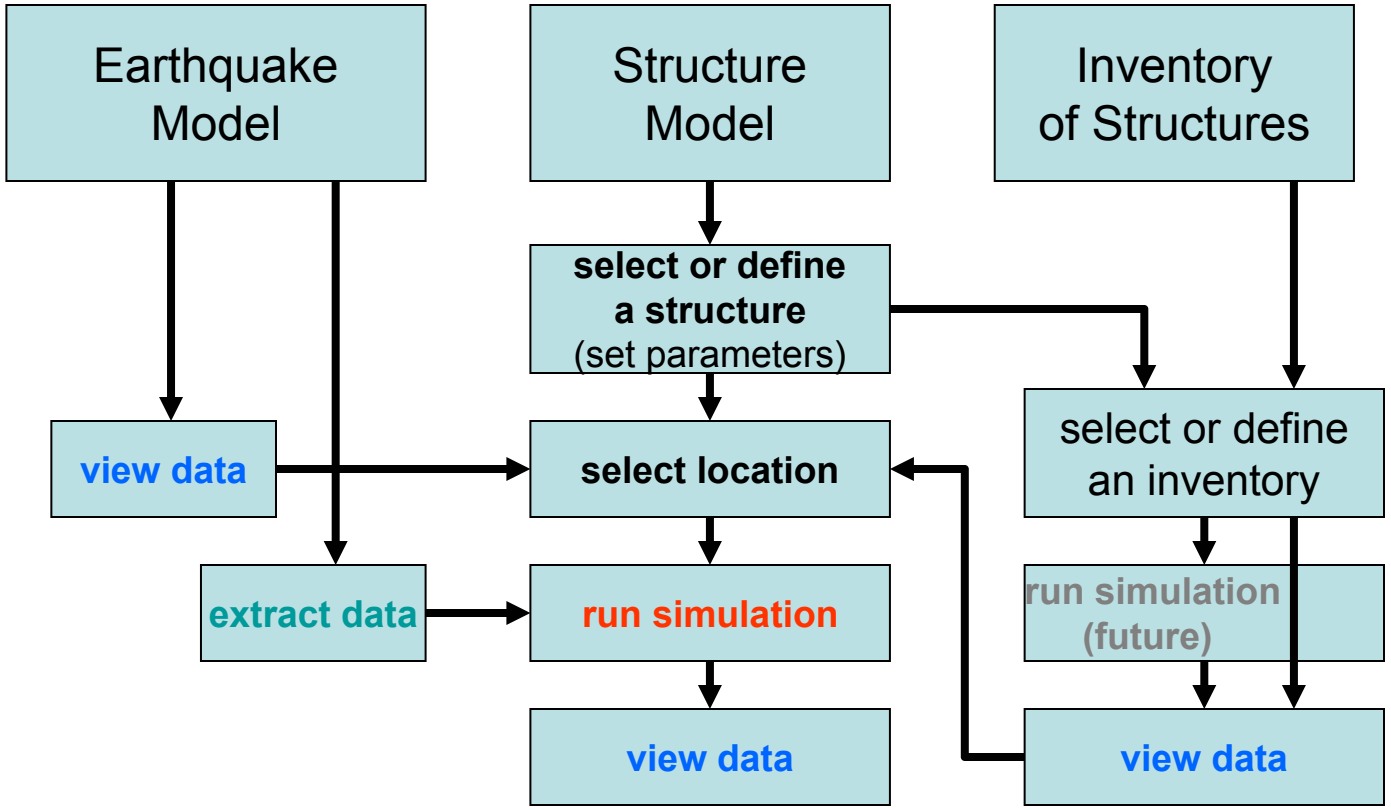
the globus project
www.globus.org

NCSA

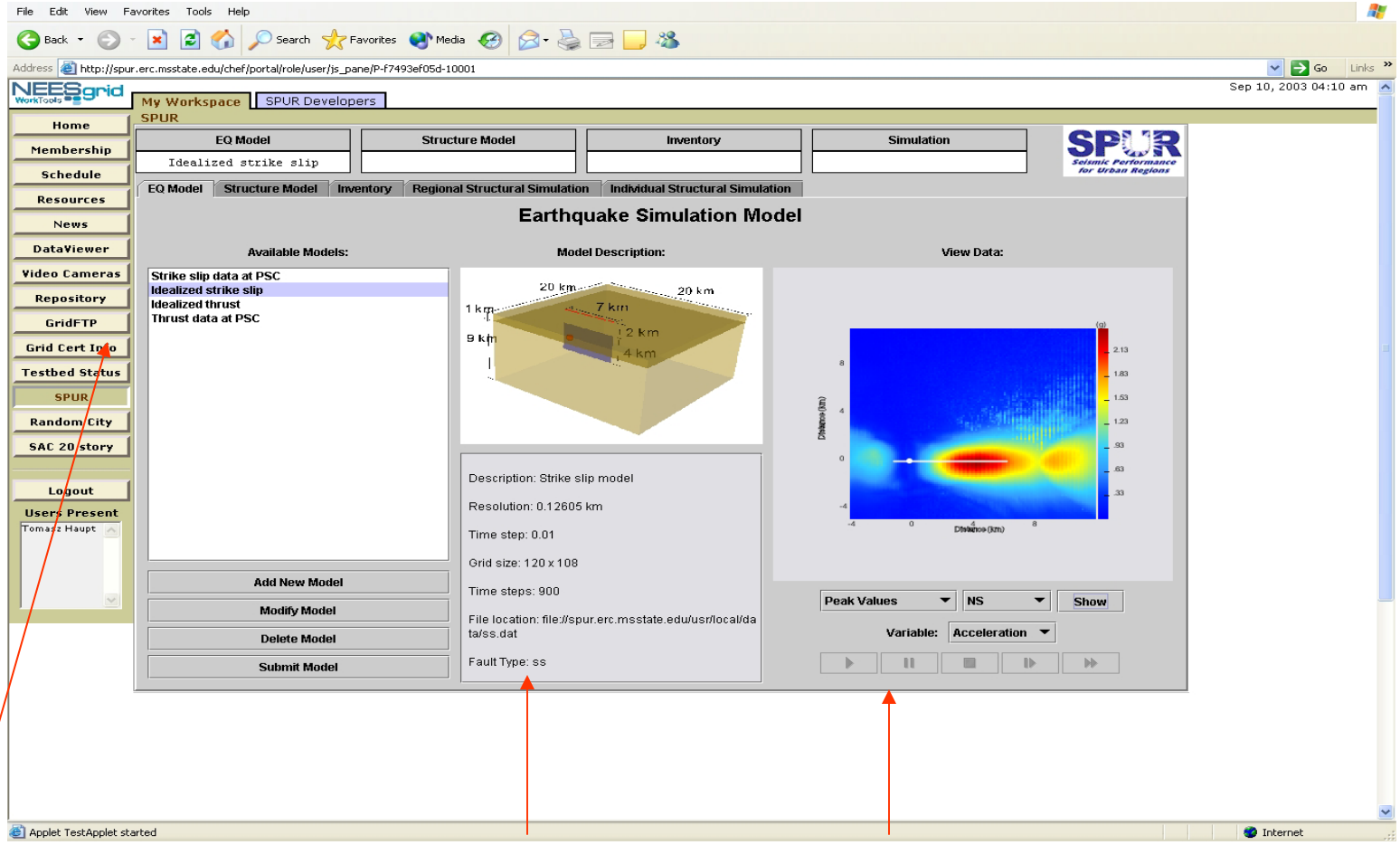
GGF

Chef
CompreHensive collaborativE Framework

# NEESport functionality



# Earthquake model

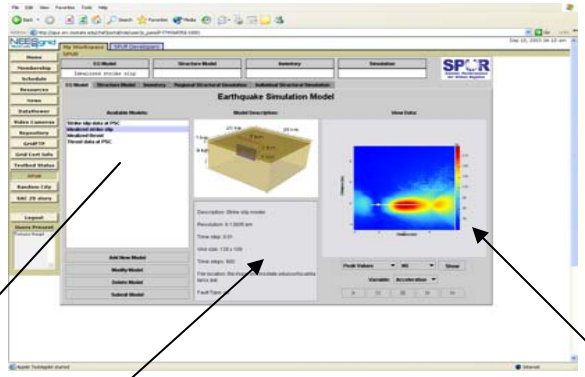



select EQ model

model description

visualize surface data

# Earthquake model (2)




logical→physical; data extraction

Ground Motion Metadata Repository Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

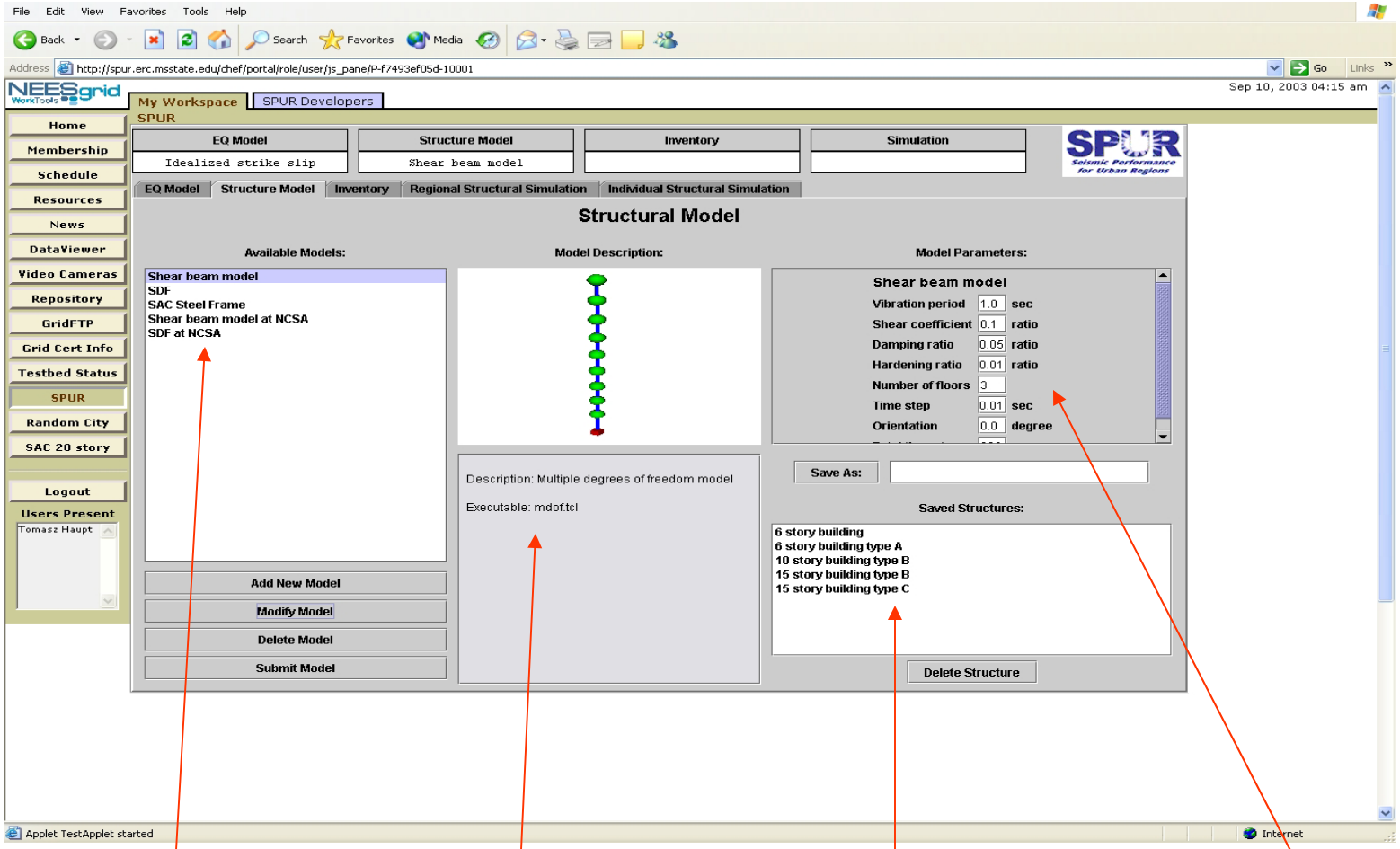
DBMS

GRAM/ MMJFS

GASS

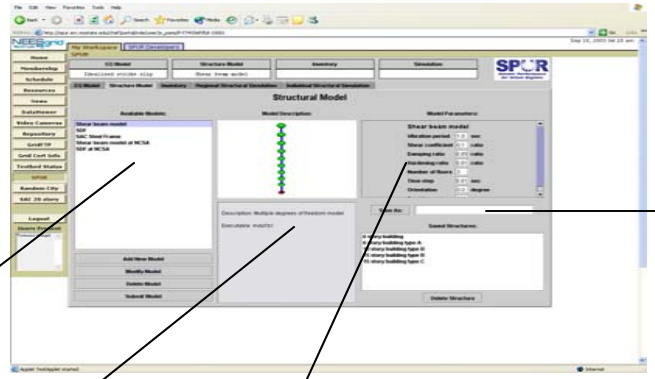
File System

# Structural Model




select structural model  model description  model instances  set model parameters

# Structural Model (2)



list

extract metadata record

extract list of model parameters

create instance of the model

Structure Models Metadata Repository Service

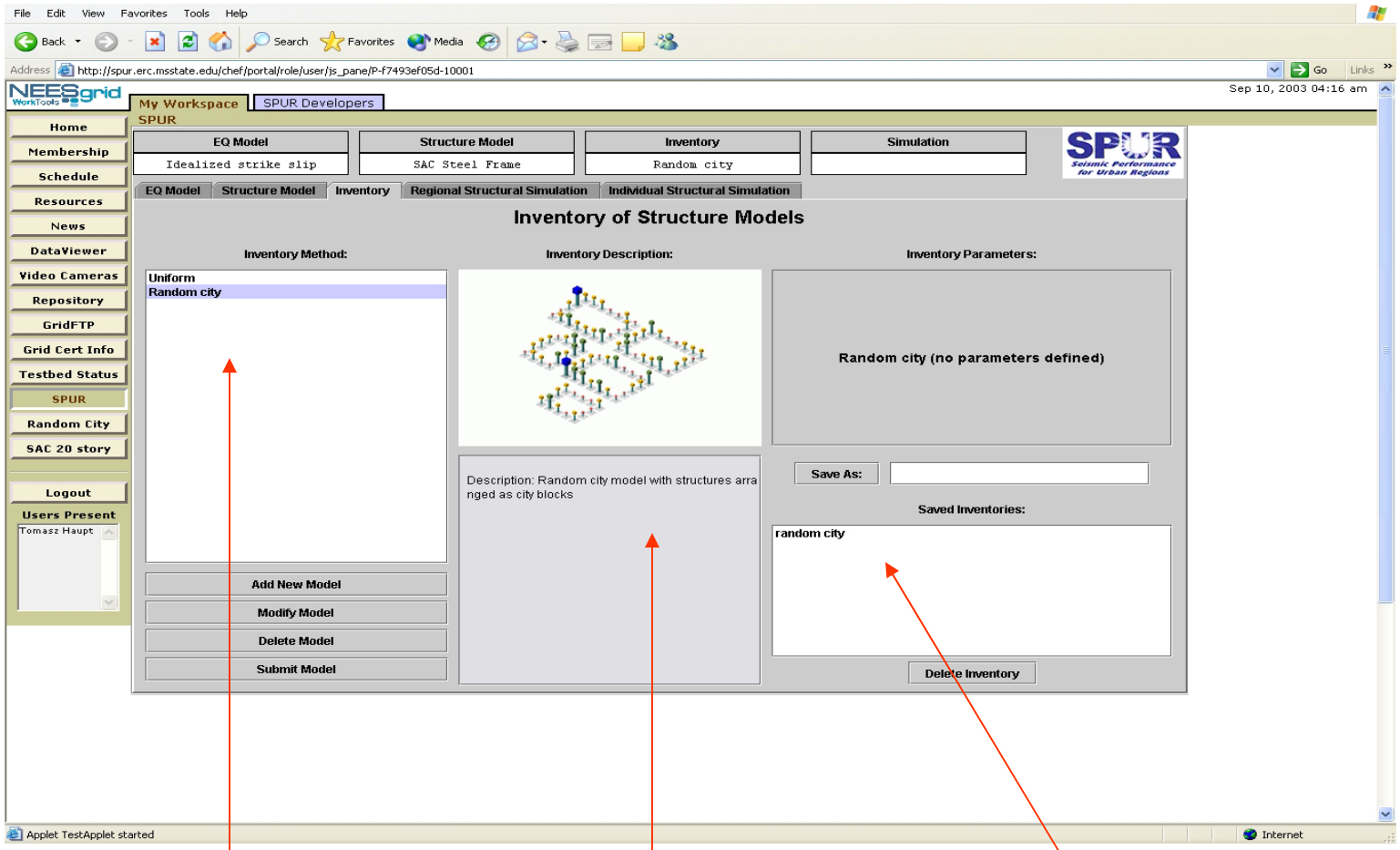
EJB

DBMS

Grid Job Descriptor (model metadata)

| |
|---|
| Application Signature and Description |
| Parameters, arguments, i/o |
| JSDL-style descriptor (for automatic generation of RSL) |
| values of parameters, run-time info (for provenance service) |

# Population Method

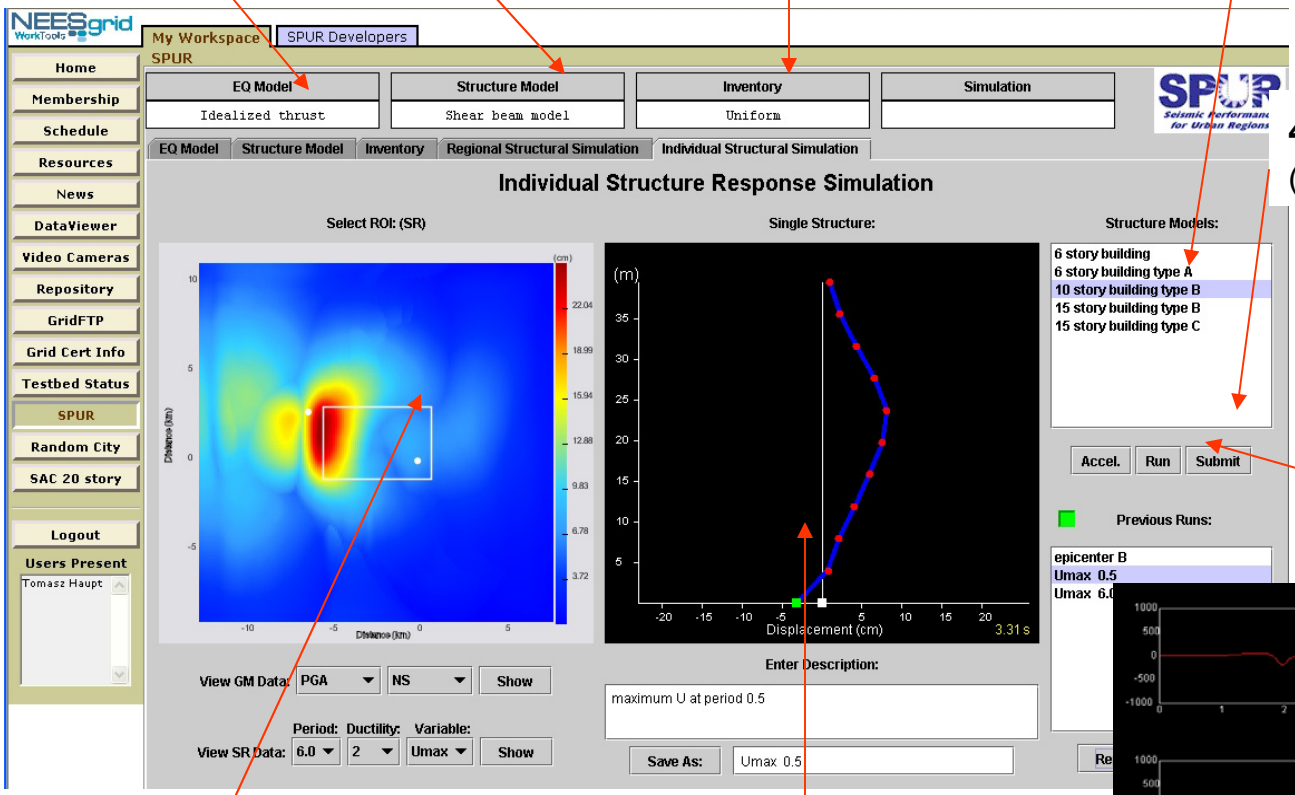



select population method  population description  population instances

# Individual Structure Response

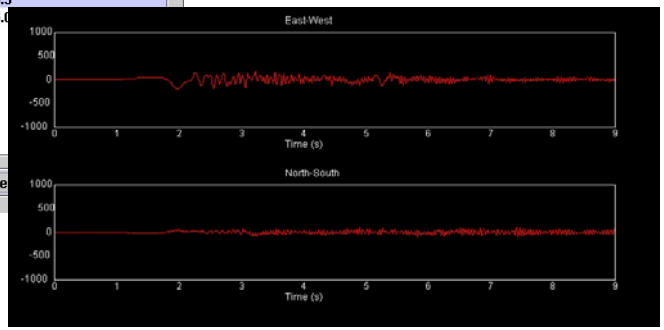
selected EQmodel  selected structure model  selected  population method

3. select structure instance




4. run openSees
(structure response simulation)

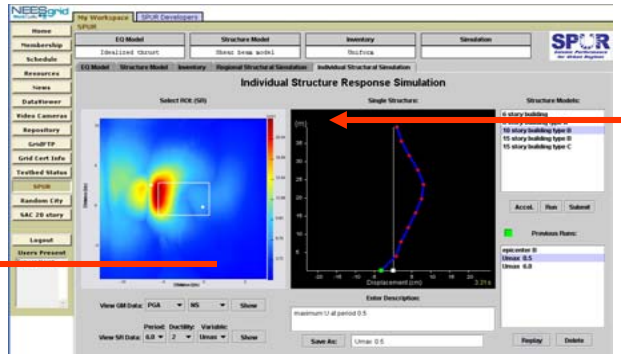
2. see acceleration history for the selected location




1. select location of structure

5. visualize results

# Individual Structure Response: Ground Motion Data




Ground Motion Metadata Repository Service

Structure Models Metadata Repository Service

Persistence & Provenance Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

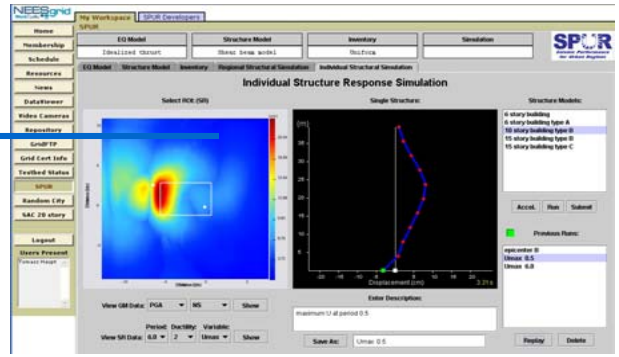
DBMS

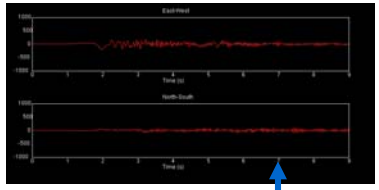
GRAM/ MMJFS

GASS

File System

# Individual Structure Response: Select Location



Ground Motion Metadata Repository Service

Structure Models Metadata Repository Service

Persistence & Provenance Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

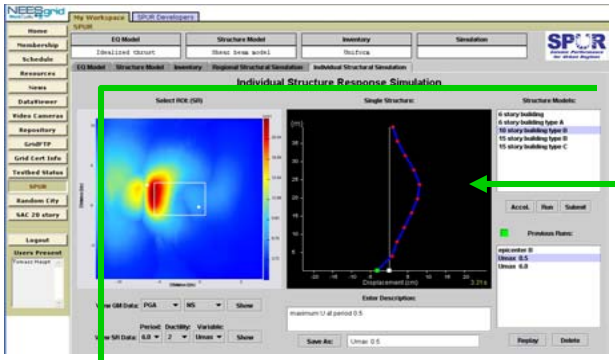
DBMS

GRAM/ MMJFS

GASS

File System

# Individual Structure Response: Select Structure & Run



Ground Motion Metadata Repository Service

Structure Models Metadata Repository Service

Persistence & Provenance Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

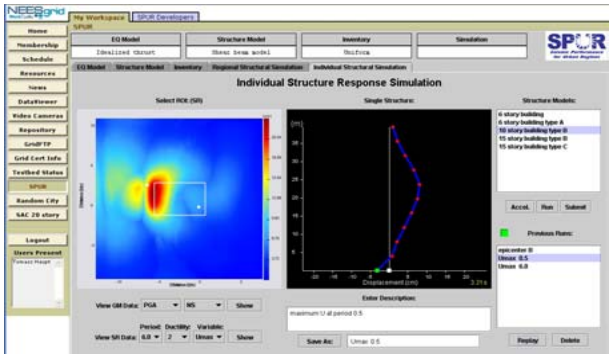
DBMS

GRAM/ MMJFS

GASS

File System

# Individual Structure Response: Replay




Ground Motion Metadata Repository Service

Structure Models Metadata Repository Service

Persistence & Provenance Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

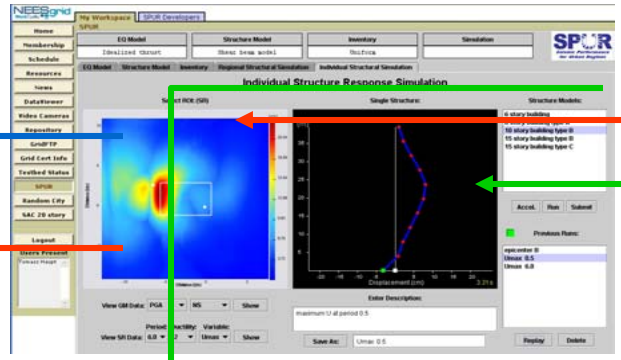
DBMS

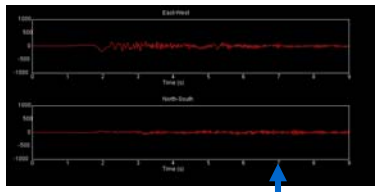
GRAM/ MMJFS

GASS

File System

# Individual Structure Response





Ground Motion Metadata Repository Service

Structure Models Metadata Repository Service

Persistence & Provenance Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

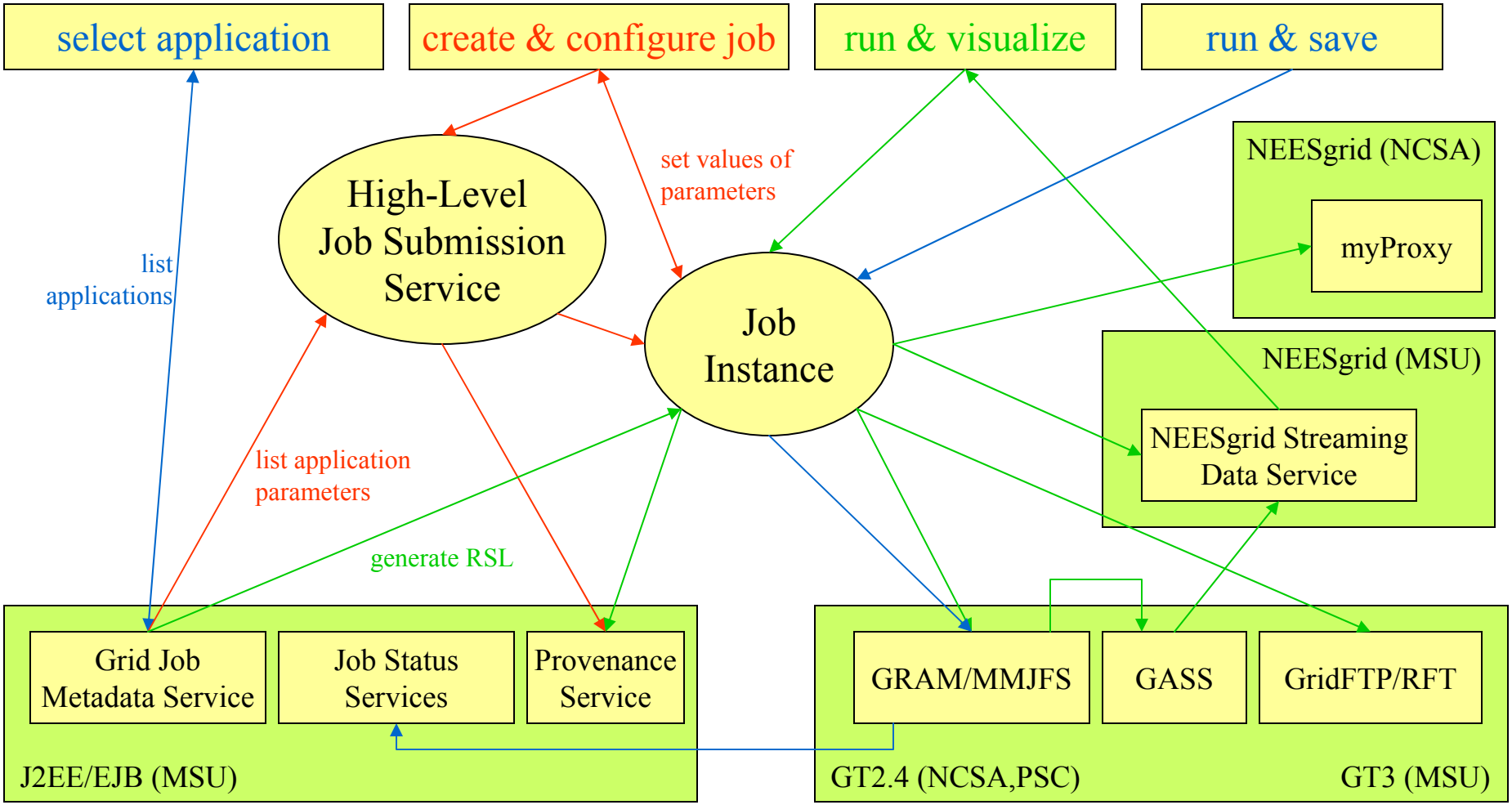
DBMS

GRAM/ MMJFS

GASS

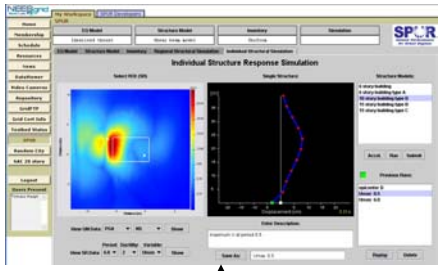
File System

# SPUR/NEESgrid Grid Solution

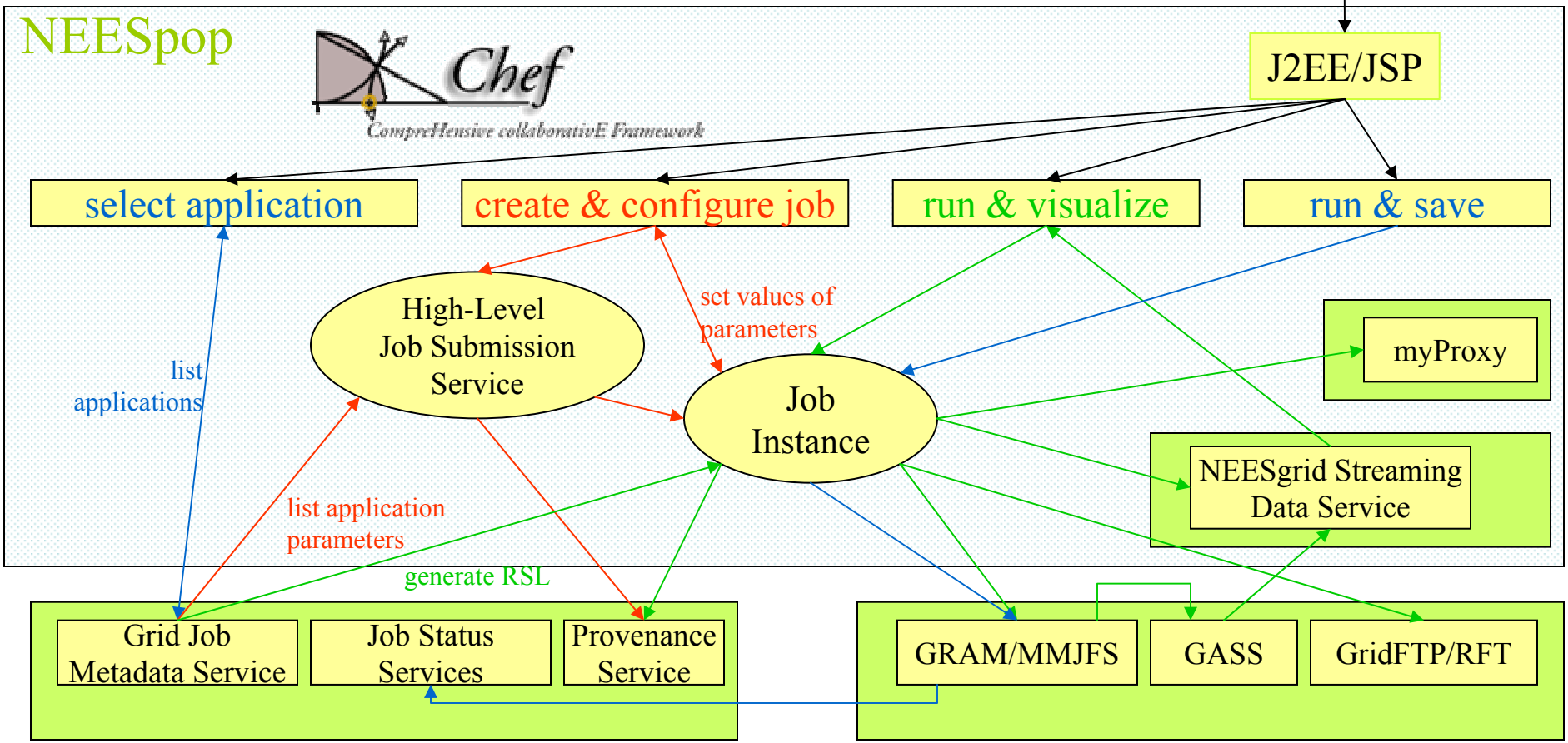


# NEESport

NEESport applet

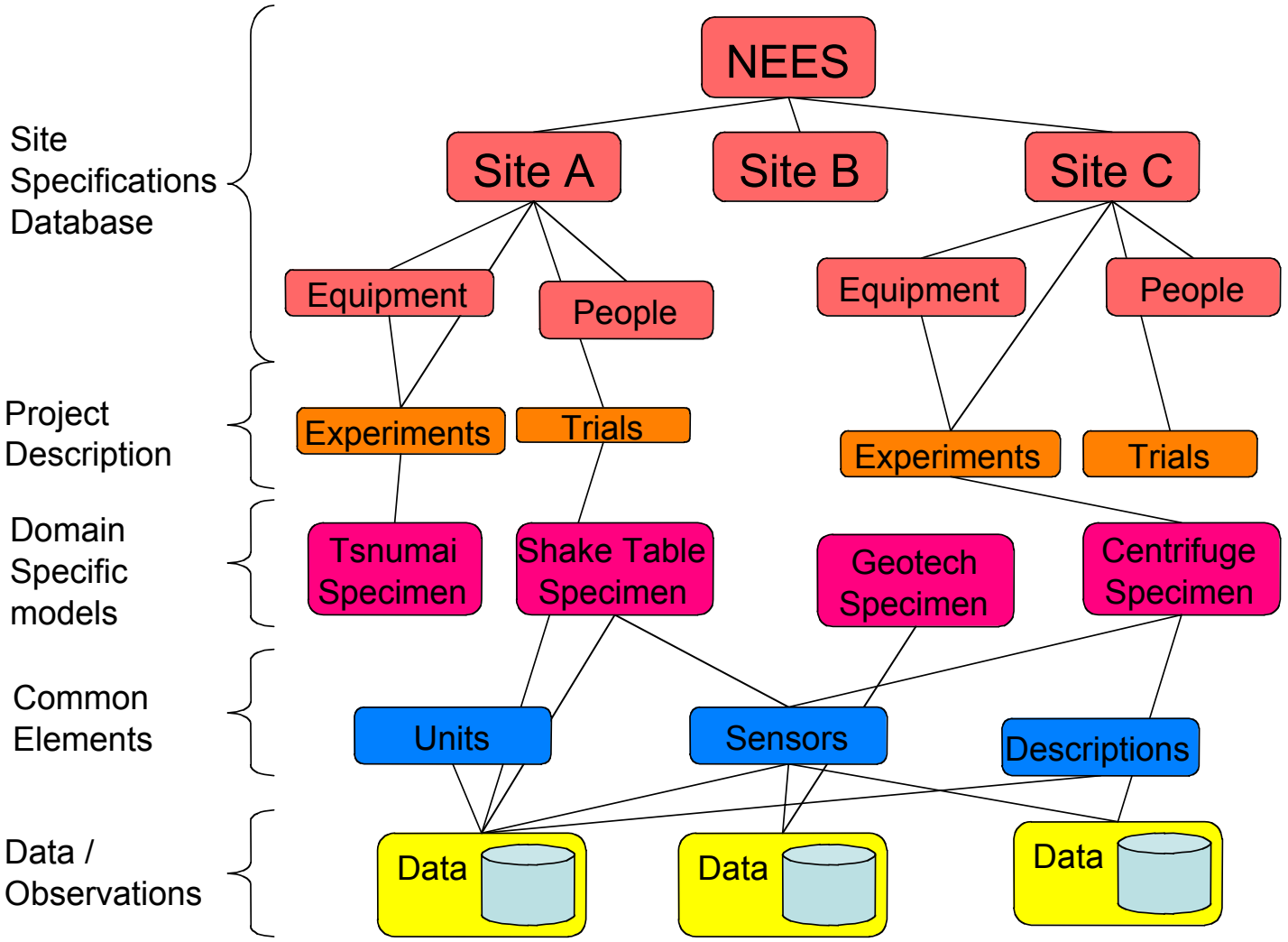


https



# NEESGrid Data Model Efforts

# Overall Data Modeling Efforts



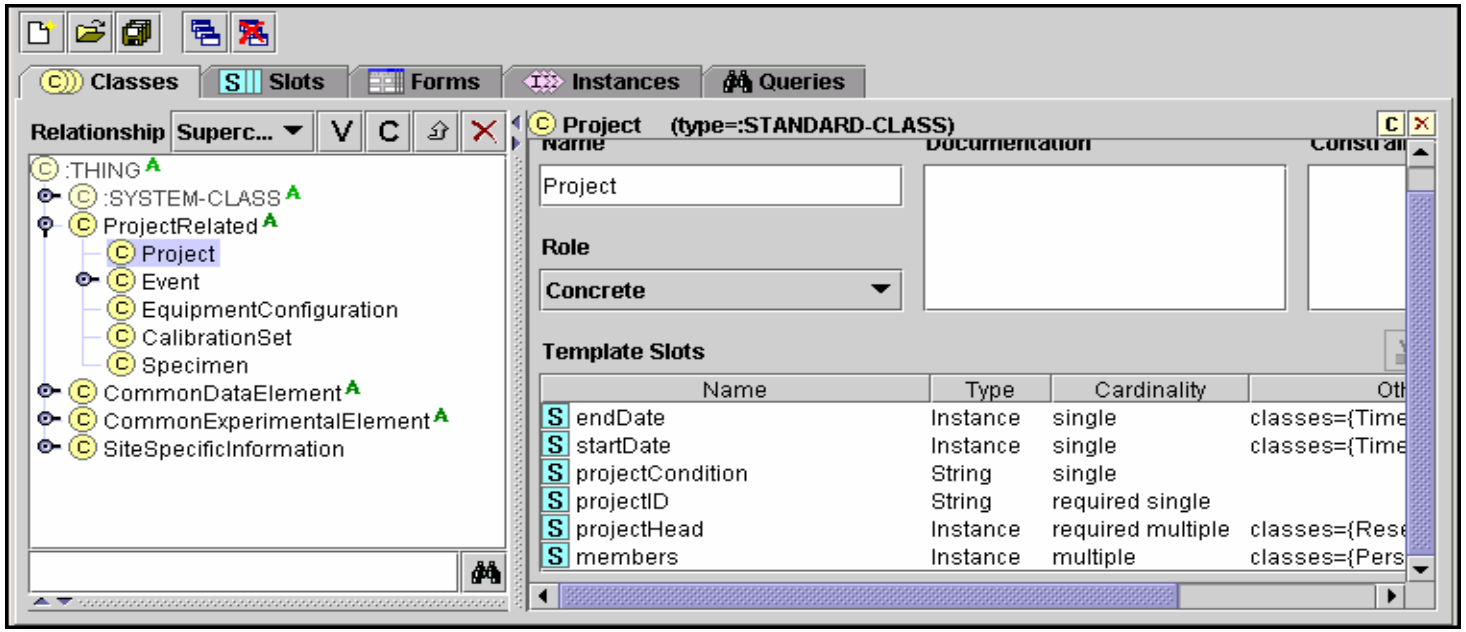
# Existing Data Model Representations

- **E-R** (Entity Relationship) Diagrams
  - *Entities*, members of an objects set
  - *Attributes*, values describing some property of an entity
  - *Relationships*, connections among one or more entity sets
- **UML's** ORM (Object Role Models)
- **XML (**Extensible Markup Language**) Schema**
  - Encoded in XML to describe document (data) structure
  - Introduces the ideas of data types, cardinality constraints
- **RDF (**Resource Description Framework**)**
  - Encoded in XML to describe resources with labeled relationships
  - More flexible than hierarchical organizations
  - Extensible: multiple RDF schemes can be combined
- **OWL (**Web Ontology Language**)**
  - Encoded in XML to describe classes and relations
  - Part of the Semantic Web Activity

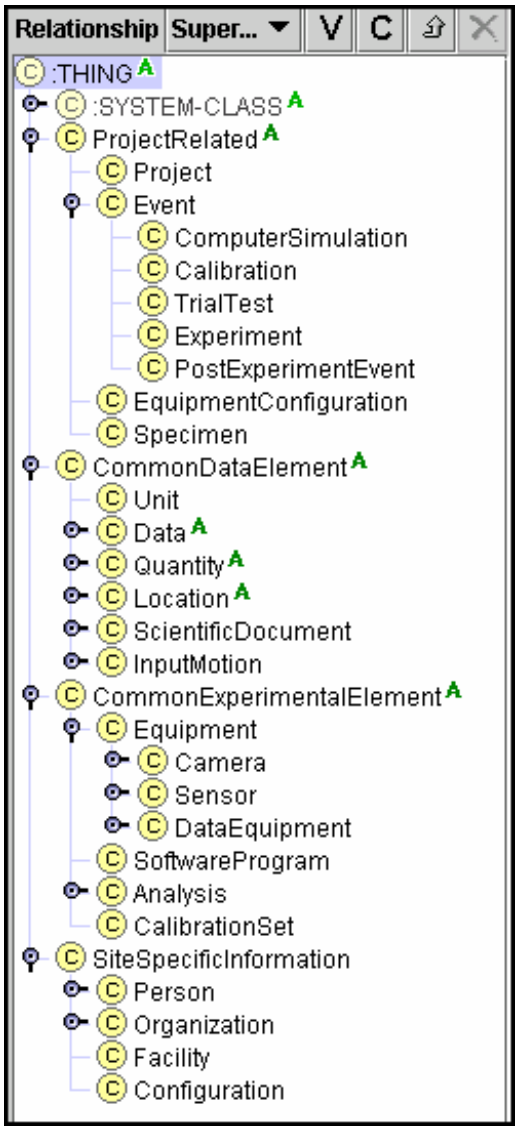
# Protégé-2000 (http://protege.stanford.edu)

Open Source Ontology Modeling Tool (with many Plugins)

- **A tool** which allows the user to construct a domain ontology
- **A platform** which can be extended with graphical widgets for tables, diagrams, animation components to access other knowledge-based systems
- **A library** which other applications can use to access knowledge bases
- Produces schemas in various data model representations



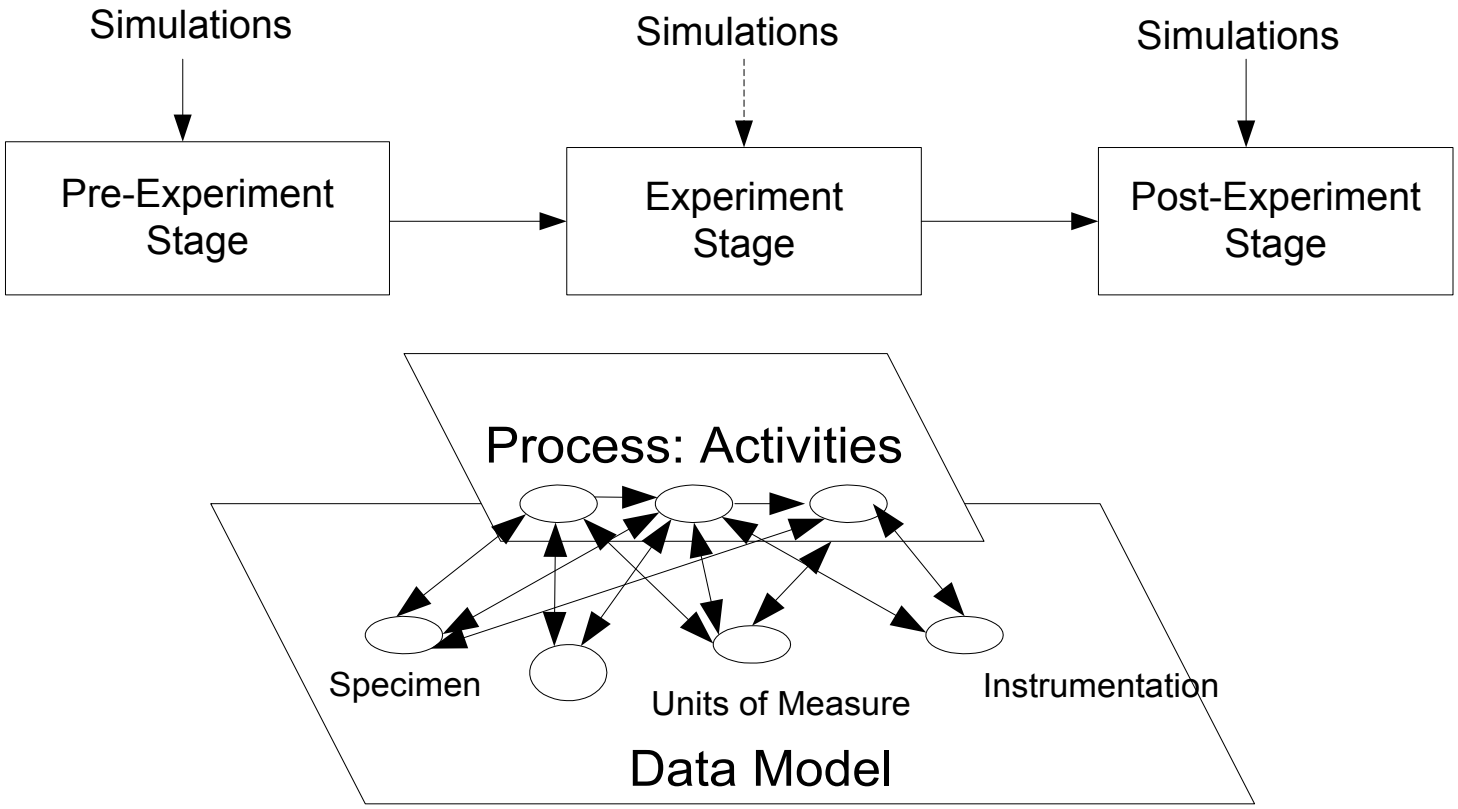
# Prototype Data Model




- Tool: Protégé-2000
- Four groups of classes:
  - ProjectRelated
  - SiteSpecificInformation
  - CommonDataElement
  - CommonExperimentalElement
- Project-centric
- Shake table test (Stanford)
- Geotechnical / centrifuge tests (USC)
- Tsunami (Oregon State)

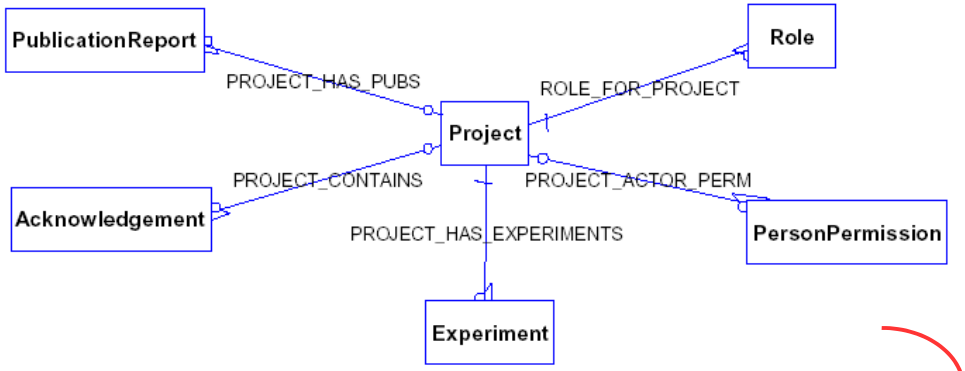
# Observations

- Pre-experiment and post-experiment data could as valuable as the actual experiment itself
- Computer simulations play a significant role towards the design of an experiment as well as for post-event investigations



# Project Entity – OrSt Model

| NEES Data Model, Oregon State University | Version: 1.0 |
|---|---|
| Data Model Documentation | Date: 03/10/2003 |



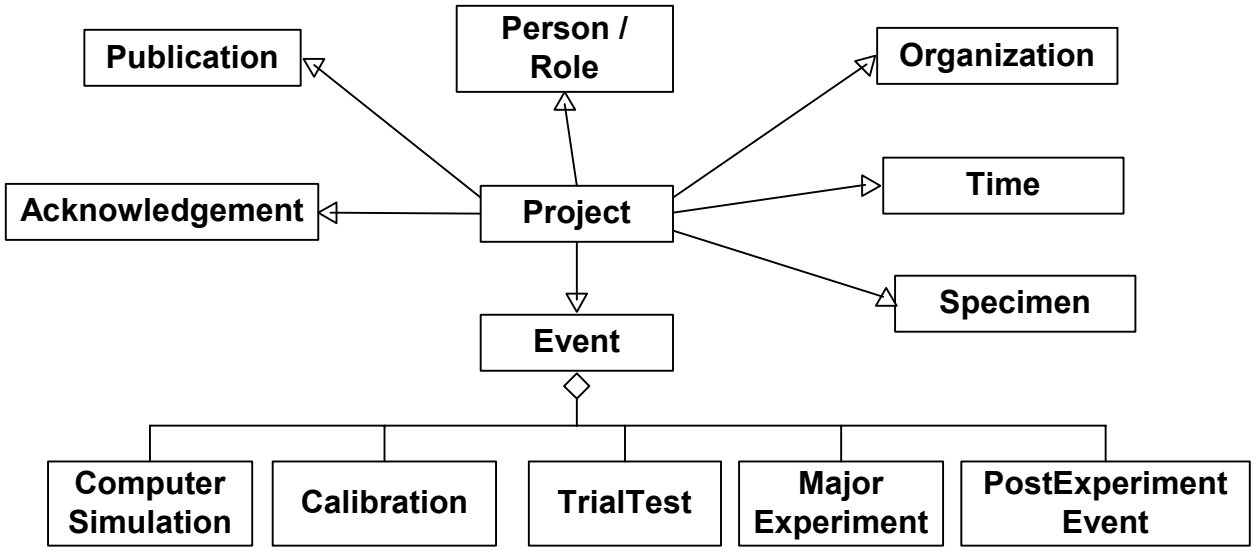

| Field Name | Description |
|---|---|
| Project ID | |
| Conditions-And-Limitations | |
| NEES-Code | |
| Lab.-Code | |
| Title | |
| Short-Description | |
| Long-Description | |
| Objectives | |
| Keywords | |
| Start-Date | |
| End-Date | |
| Create-Date | |

**Project Fields**

| Field Name | Description |
|---|---|
| projectID | Primary key, or unique identifier for this Project. |
| specialConditions | Describes any special conditions, limitations, or features associated with the Project as a whole. |
| NEESCode | A reference code assigned by the NEES Consortium for the Project. |
| labCode | A reference code assigned by the specific lab for the Project. |
| title | Brief title for the Project. Since this will be used to identify the Project in lists and menus, we recommend limiting the title to approximately 20 characters. |
| shortDescription | Short, "display-style" description of the Project. |
| longDescription | More detailed description of the Project, its goals, and other high-level information. |
| startDate | Start date for the Project. Will always be on or before the startDate of the earliest Experiment belonging to the Project. |
| endDate | End date for the Project. Will always be on or after the endDate of the latest Experiment belonging to the Project. |
| createDate | Date when the Project record was added to the database. |

| Table Relationship | Description |
|---|---|
| Project to PublicationReport | A Project will yield zero or more Publications/Reports (refer to Section 3.11 for information on the PublicationReport table). Publications/Reports that refer to just a particular Experiment, rather than the Project as a whole, are associated with that Experiment instead (refer to Section 3.2 for information on the Experiment table). |
| Project to Role | Role of a particular individual with respect to the Project. |
| Project to Acknowledgement | A Project will be associated with zero or more Acknowledgements (refer to Section 3.10 for information on the Acknowledgement table). |
| Project to PersonPermission | A Project may assign certain permissions to particular individuals, which will last throughout the Project (e.g., the PI may designate who has the authority to modify Project information in the database). That association is made through this relationship (refer to Section 3.9 for information on the PersonPermission table). |
| Project to Experiment | A Project includes zero or more Experiments (refer to Section **3.2** for information on the Experiment table). |

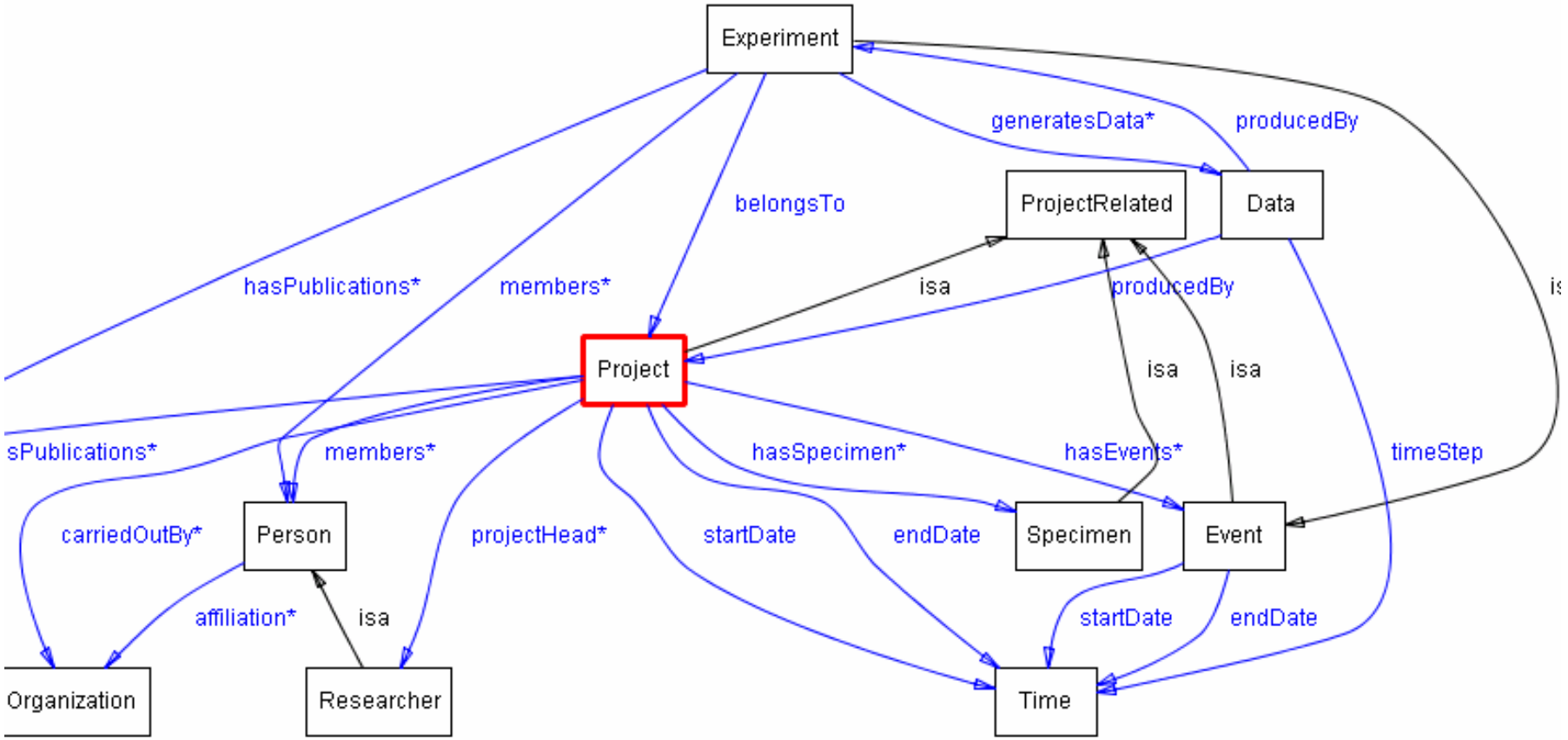
# Project Entity – Revised Model




## Key Additions to OrSt Model:

- Project has many events, which categorized in five types
- All the events have trials and versions
- Project deals with certain specimen; but specimen modeling varies widely: domain dependent, project dependent, experiment dependent

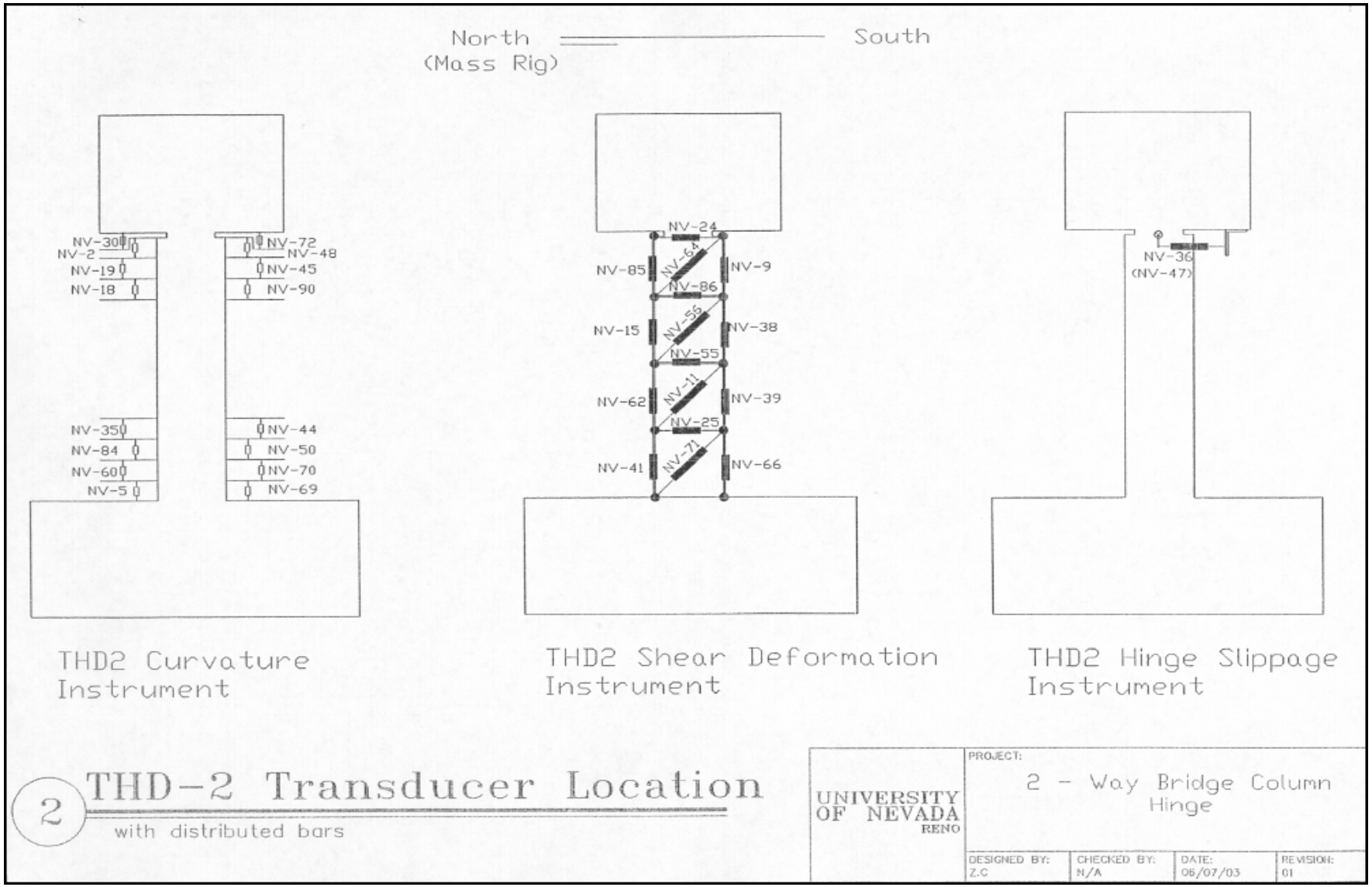
# Project Model (generated by Ontoviz)



# Specimen Modeling

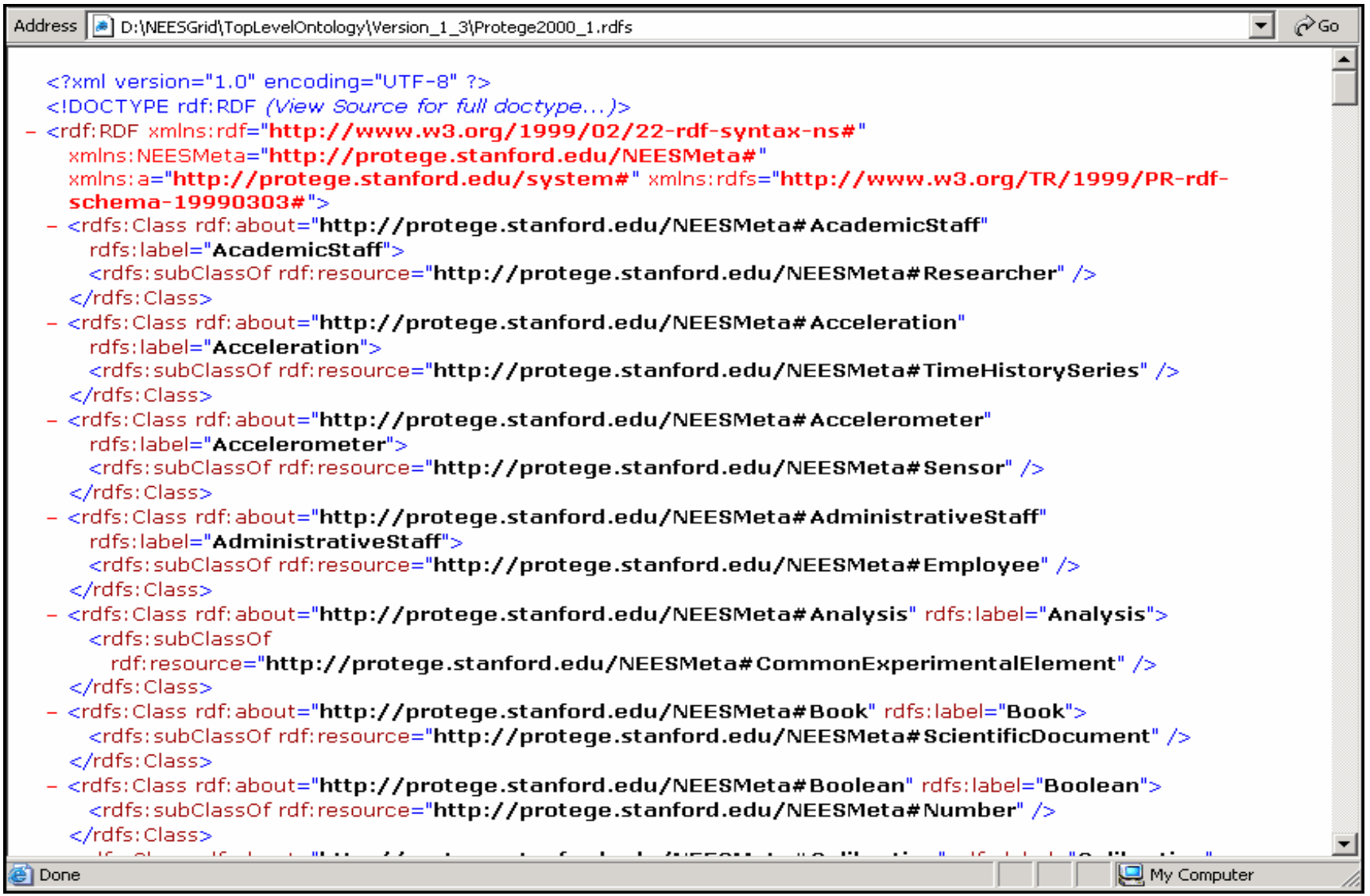
- Universal modeling of specimen for all experiments is very difficult if not impossible
- Goal is to provide ways to archive the data and information on the project and the experiment
- Basic formats and desirable features: CAD drawings; scratch drawings and notes; photos; narrative description; electronic notebook; linkage of drawings, sensor locations to data, etc..

# Drawings Indicating Sensor Locations




**Courtesy of Gokhan Pekcan, Patrick Laplace**

# Backend – RDF (Protégé output)

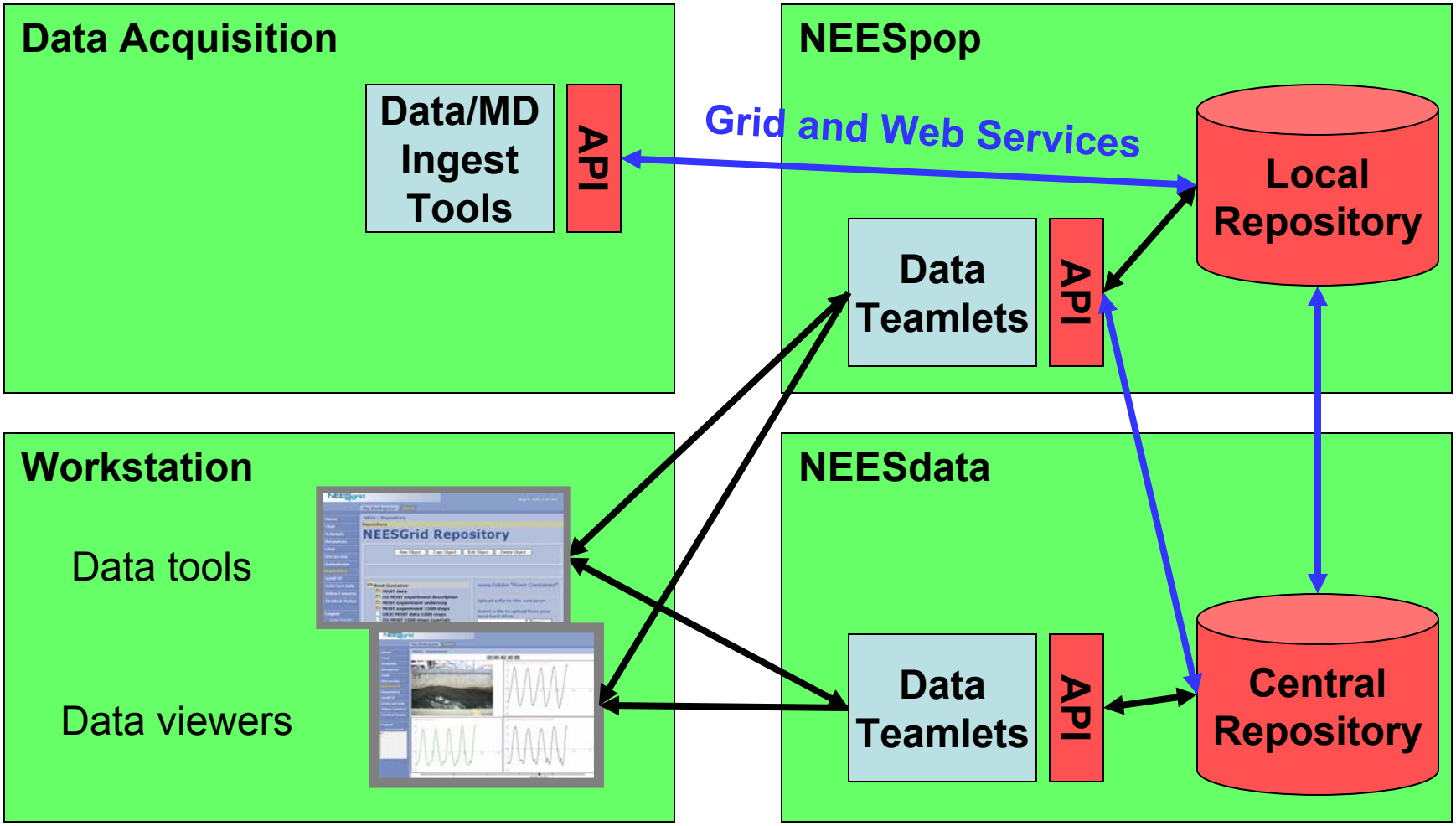


# NEESGrid Data Technologies

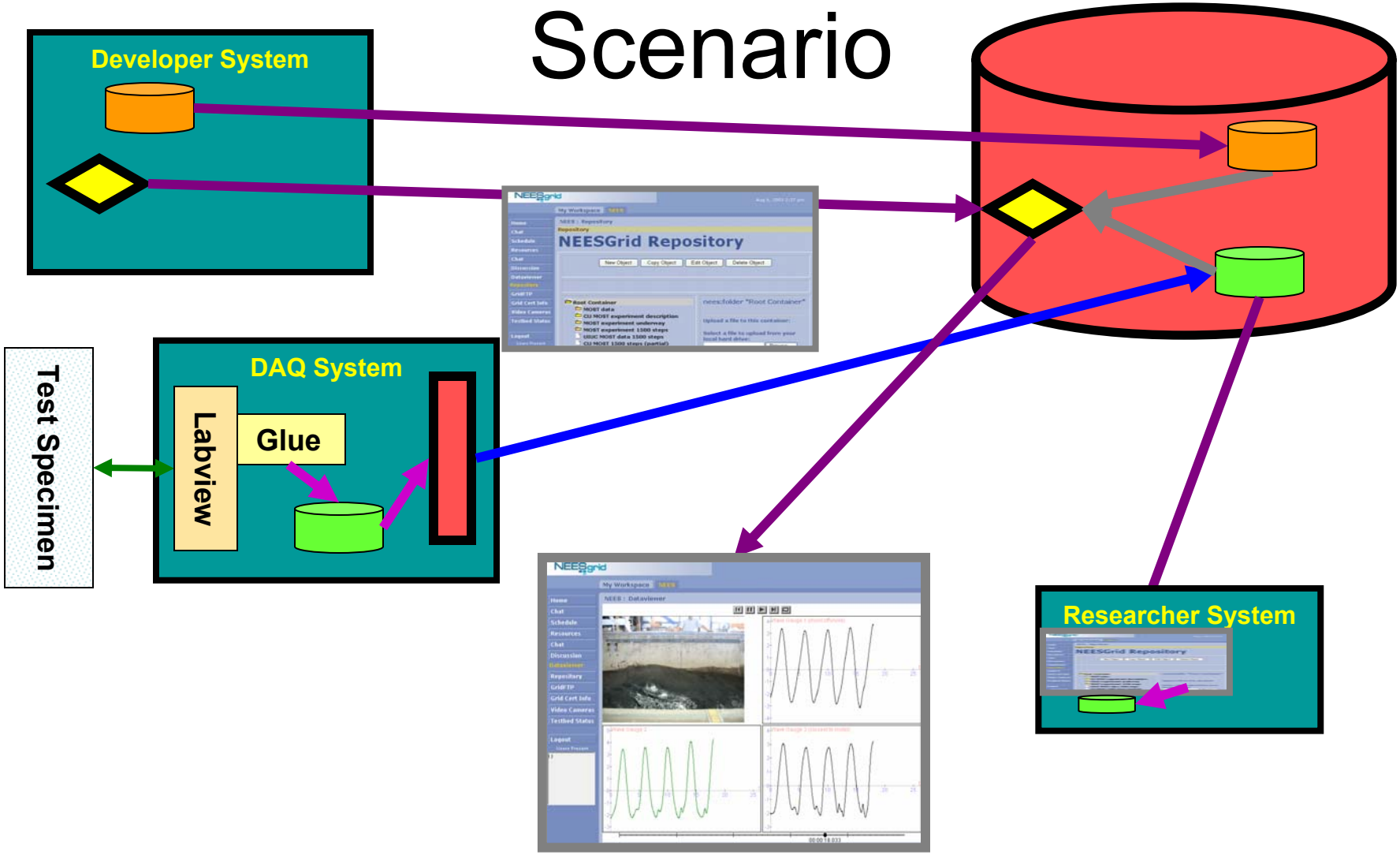
# NEESgrid Data – Core Elements

- Local Repository
- Central Repository
- JAVA APIs – Run locally on the same system as a repository or over OGSA Web Services
  - NEES File Management Services
  - NEES Meta Data Services
- Data Viewers
  - Streaming (numeric, X/Y graph)
  - Stored (X/Y graph, 2-D structure, video)

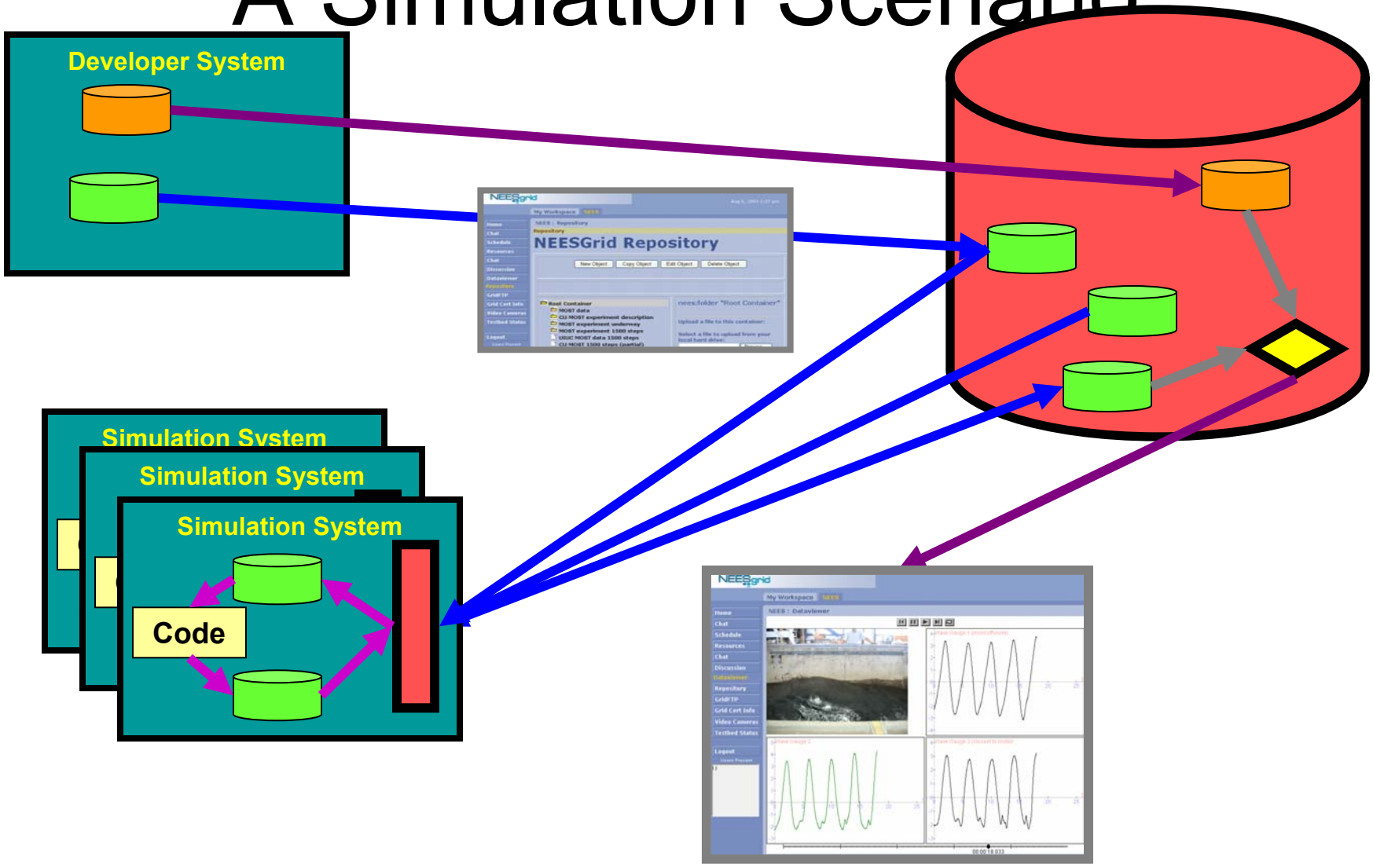
# Core Elements



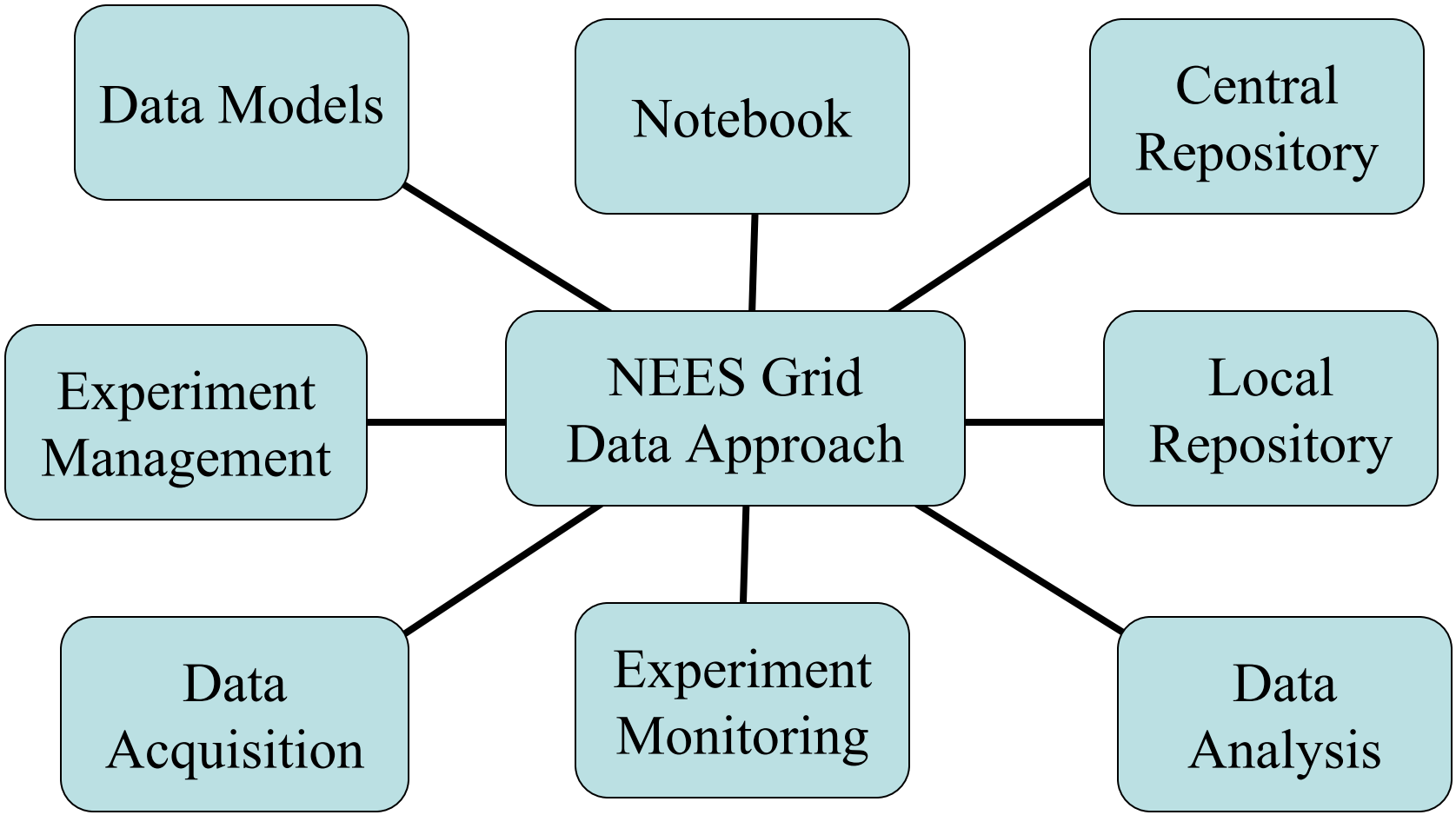
# A Simple Experimental Scenario



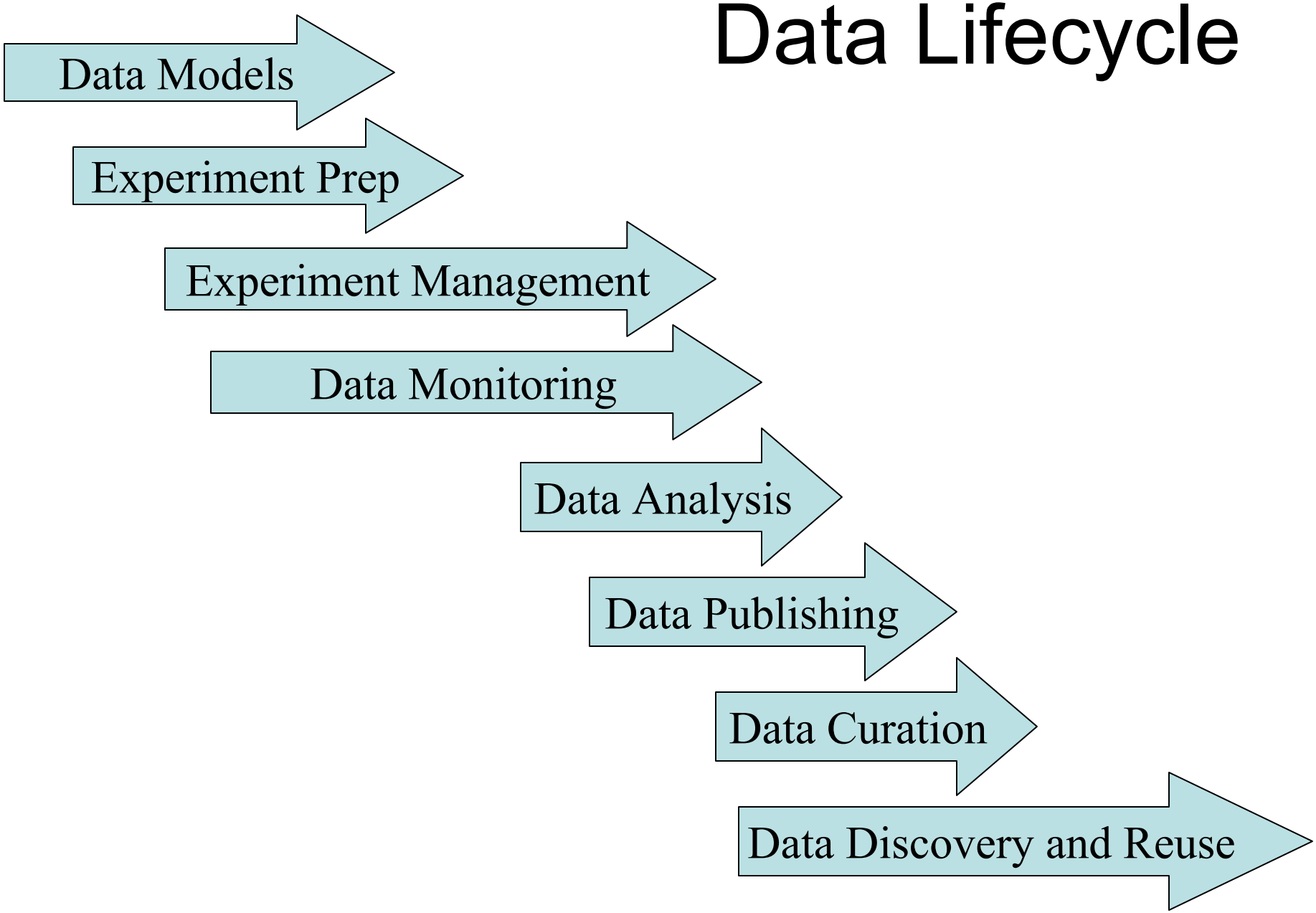
# A Simulation Scenario



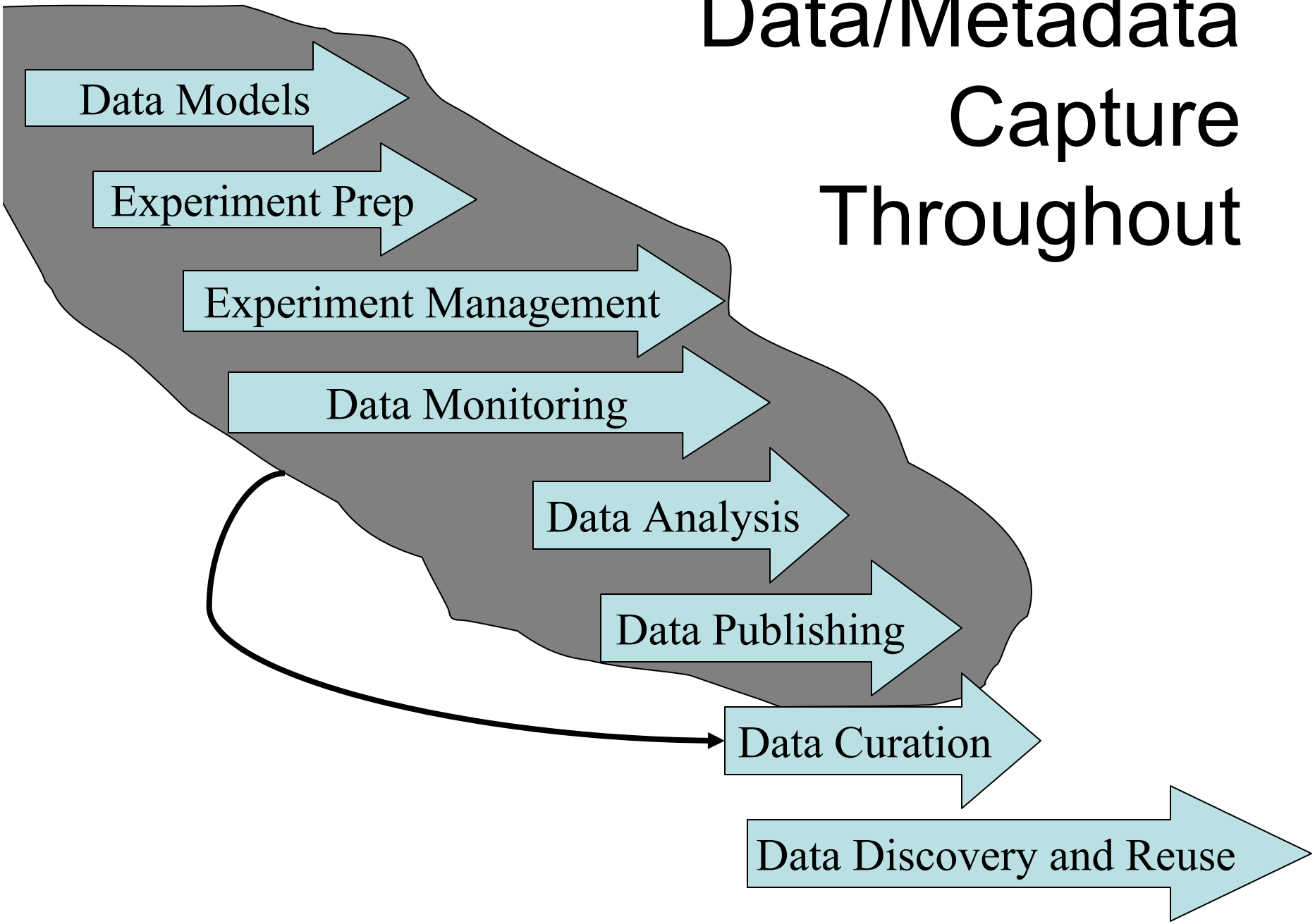
# Boxology


Data Models
Notebook
Central Repository
Experiment Management
NEES Grid Data Approach
Local Repository
Data Acquisition
Experiment Monitoring
Data Analysis

# Data Lifecycle

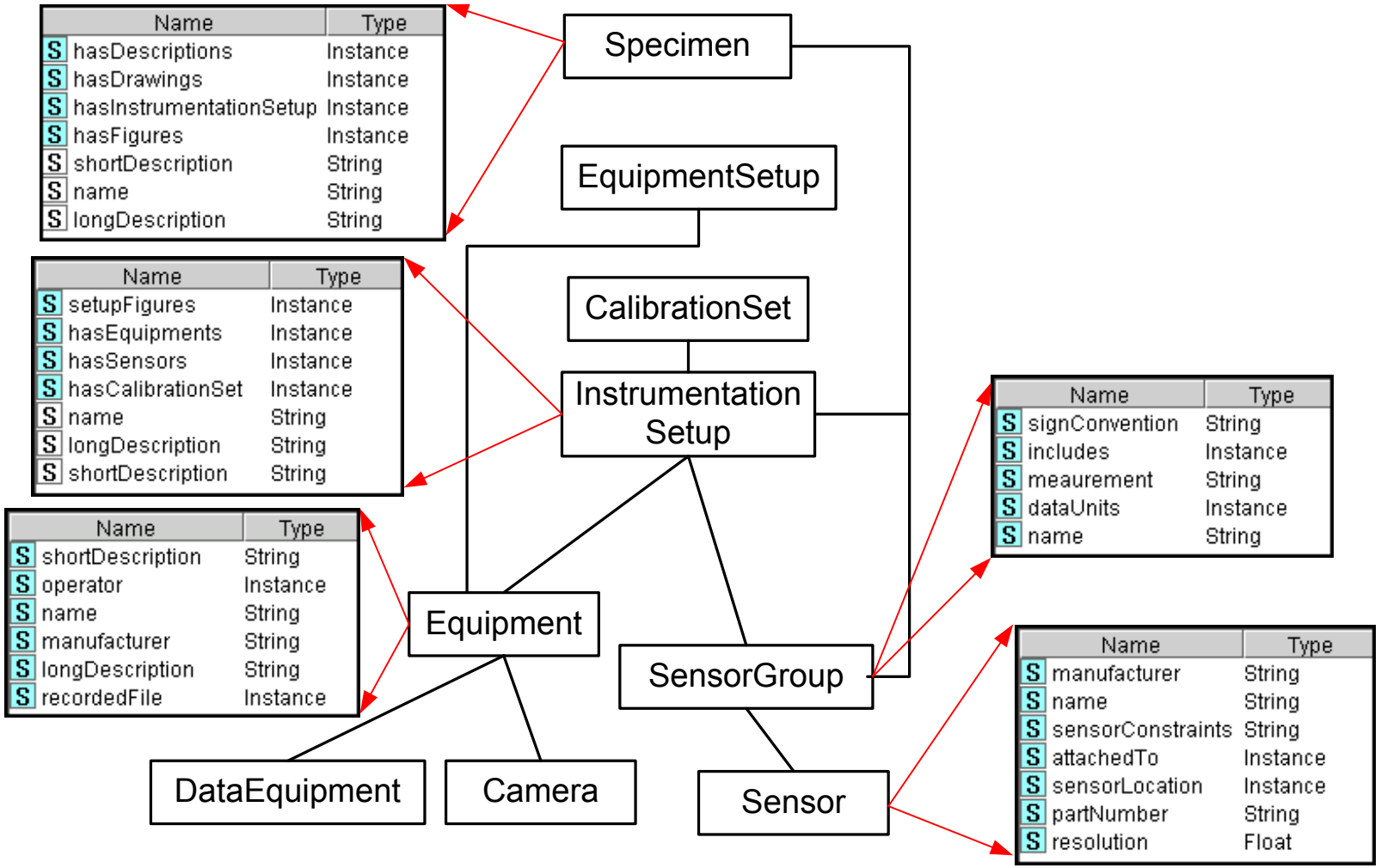


# Data/Metadata Capture Throughout

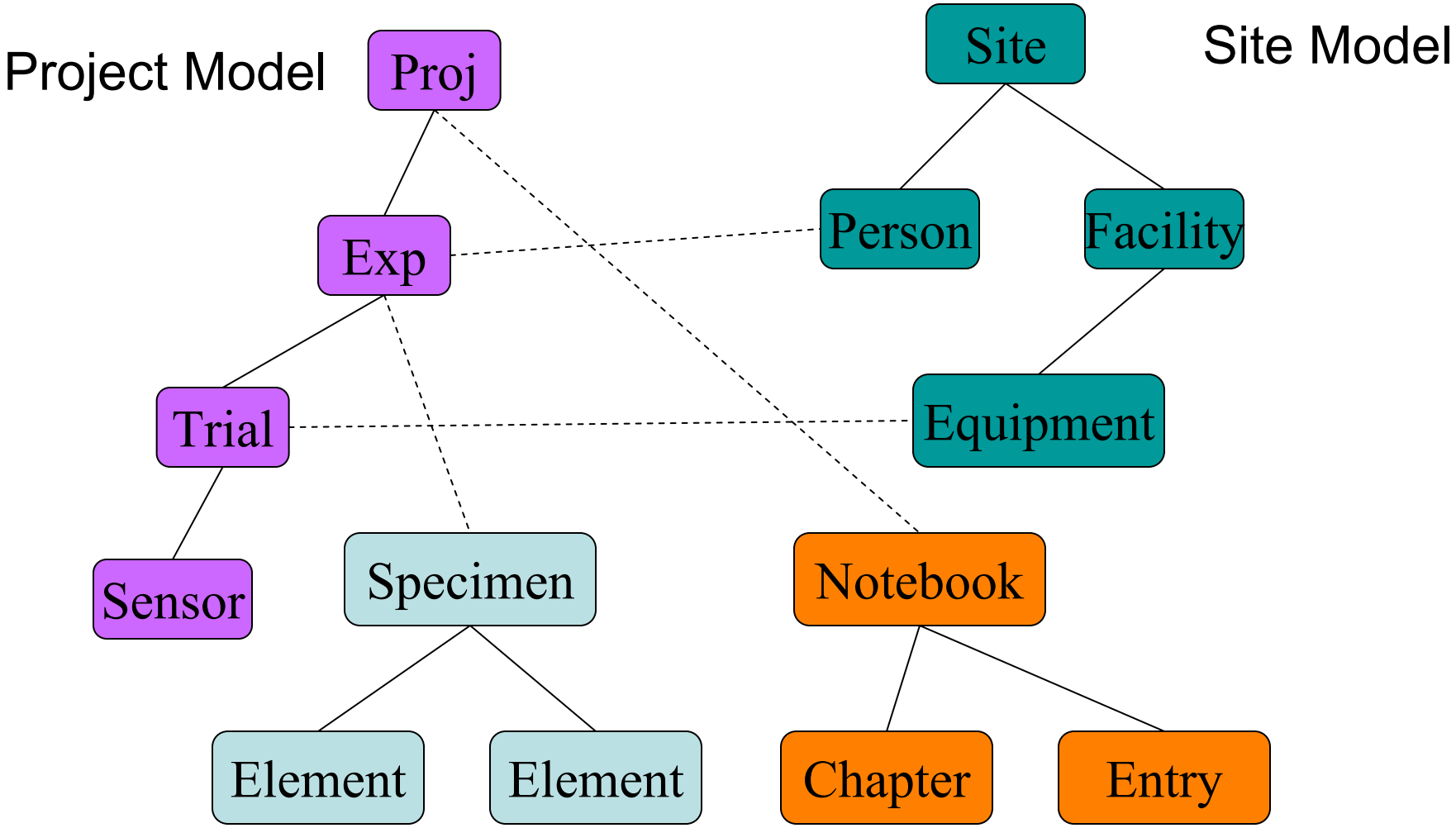


# Data Models

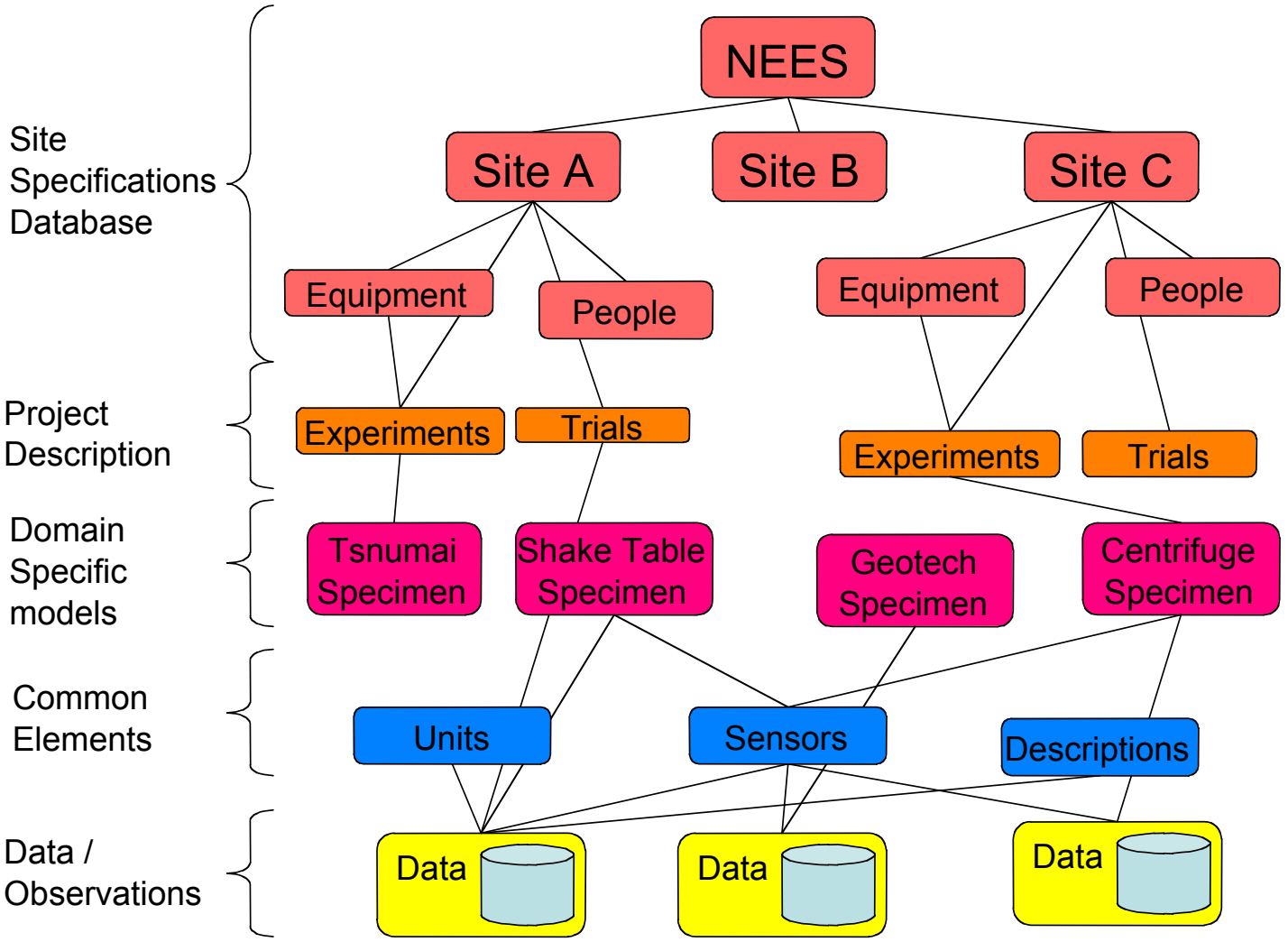
- Data models are developed in RDF
- Local repository supports multiple simultaneous data models with cross-model linkages
- Metadata browser (aka Project browser) becomes the Project Browser, Notebook Browser, Site Specification Database Browser
- Metadata browser can federate multiple sources of Metadata


Name
Type
hasDescriptions
Instance
hasDrawings
Instance
hasInstrumentationSetup
Instance
hasFigures
Instance
shortDescription
String
name
String
longDescription
String
Specimen
EquipmentSetup
CalibrationSet
Instrumentation Setup
Equipment
SensorGroup
Sensor
DataEquipment
Camera
setupFigures
Instance
hasEquipments
Instance
hasSensors
Instance
hasCalibrationSet
Instance
name
String
longDescription
String
shortDescription
String
signConvention
String
includes
Instance
meaurement
String
dataUnits
Instance
name
String
shortDescription
String
operator
Instance
name
String
manufacturer
String
longDescription
String
recordedFile
Instance
manufacturer
String
name
String
sensorConstraints
String
attachedTo
Instance
sensorLocation
Instance
partNumber
String
resolution
Float

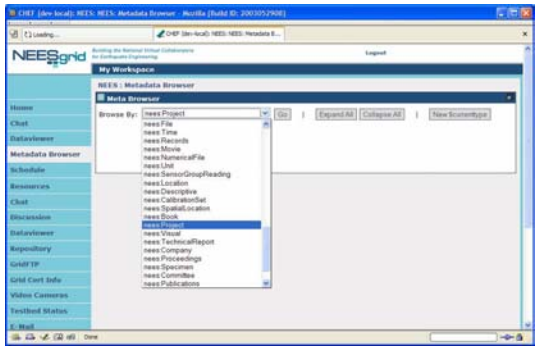
# Multiple Models



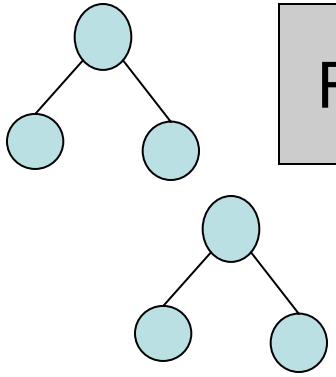
# Overall Data Modeling Efforts




**Ref. Source: Chuck Severance**



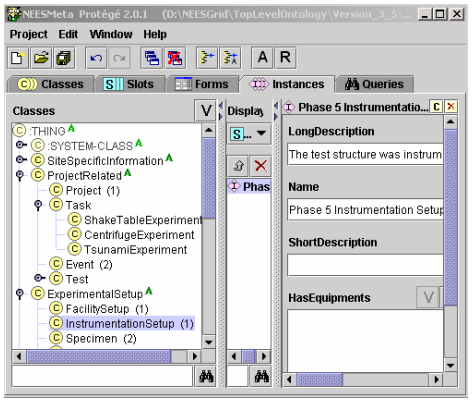
Data



RDF

Load

Models



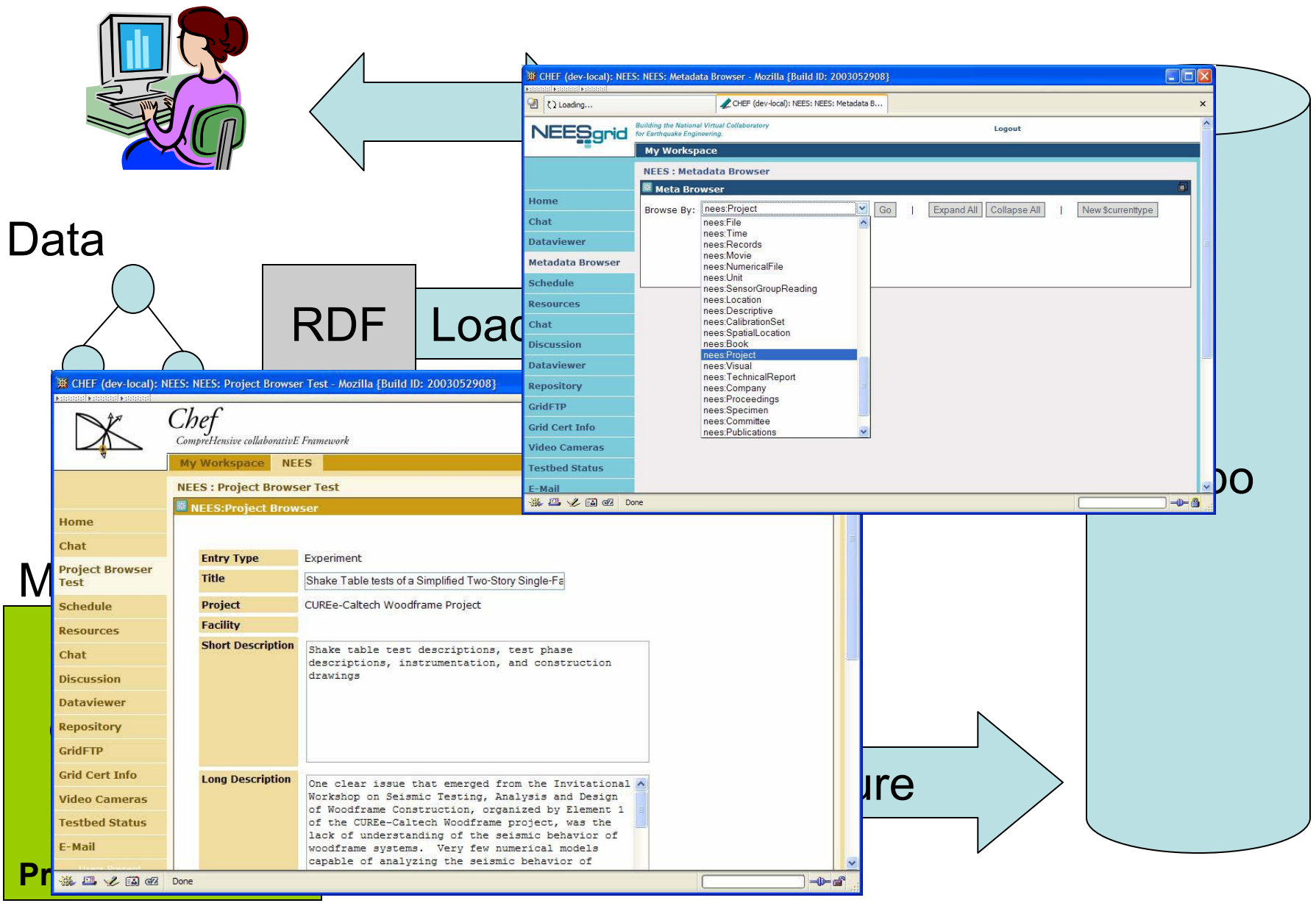

Configure

```
<owl:ObjectProperty rdf:ID="hasPublications">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Project"/>
        <owl:Class rdf:about="#Task"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Publications"/>
</owl:ObjectProperty>
```

RDF/
OWL

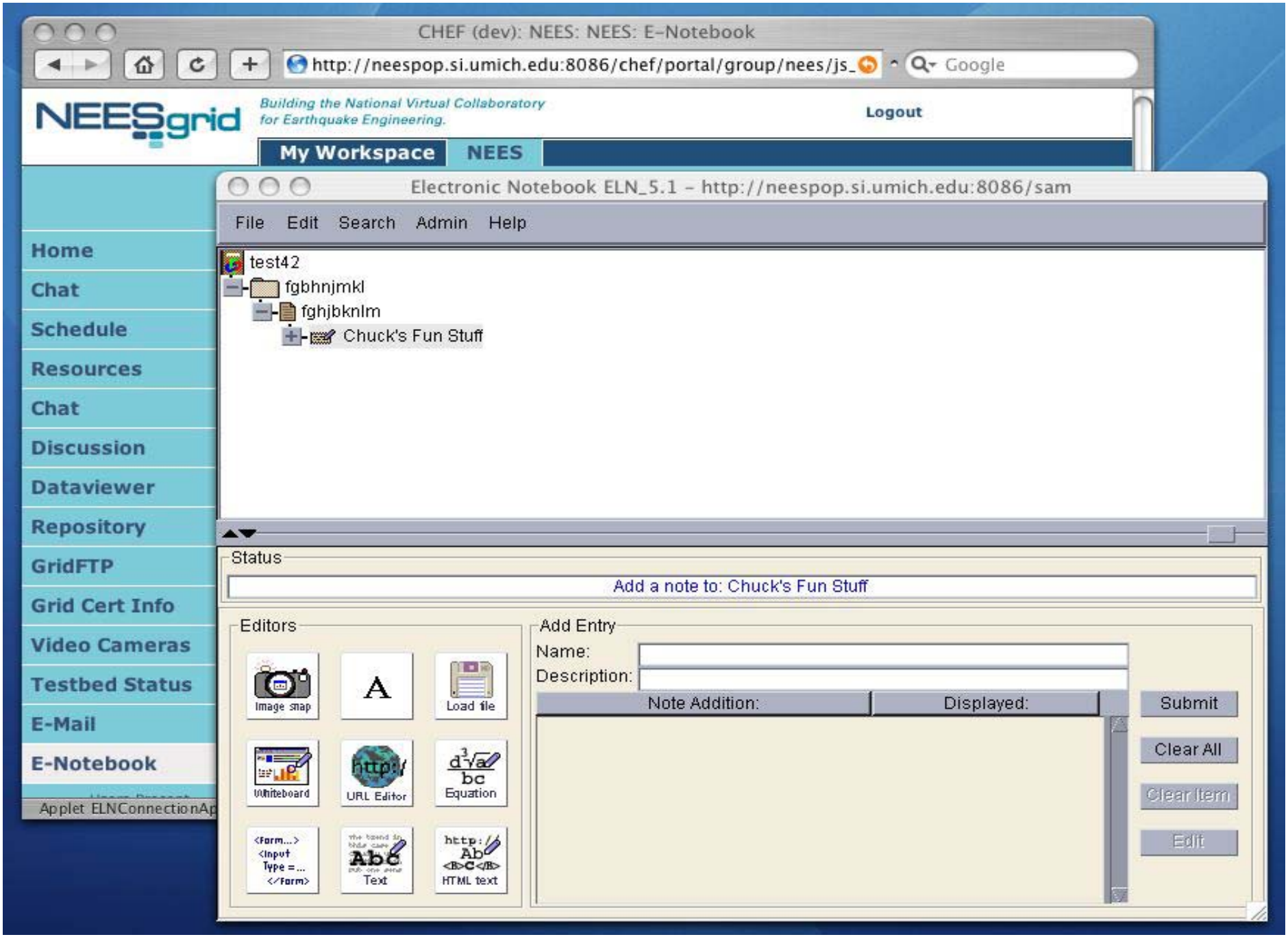
Configure

Repo

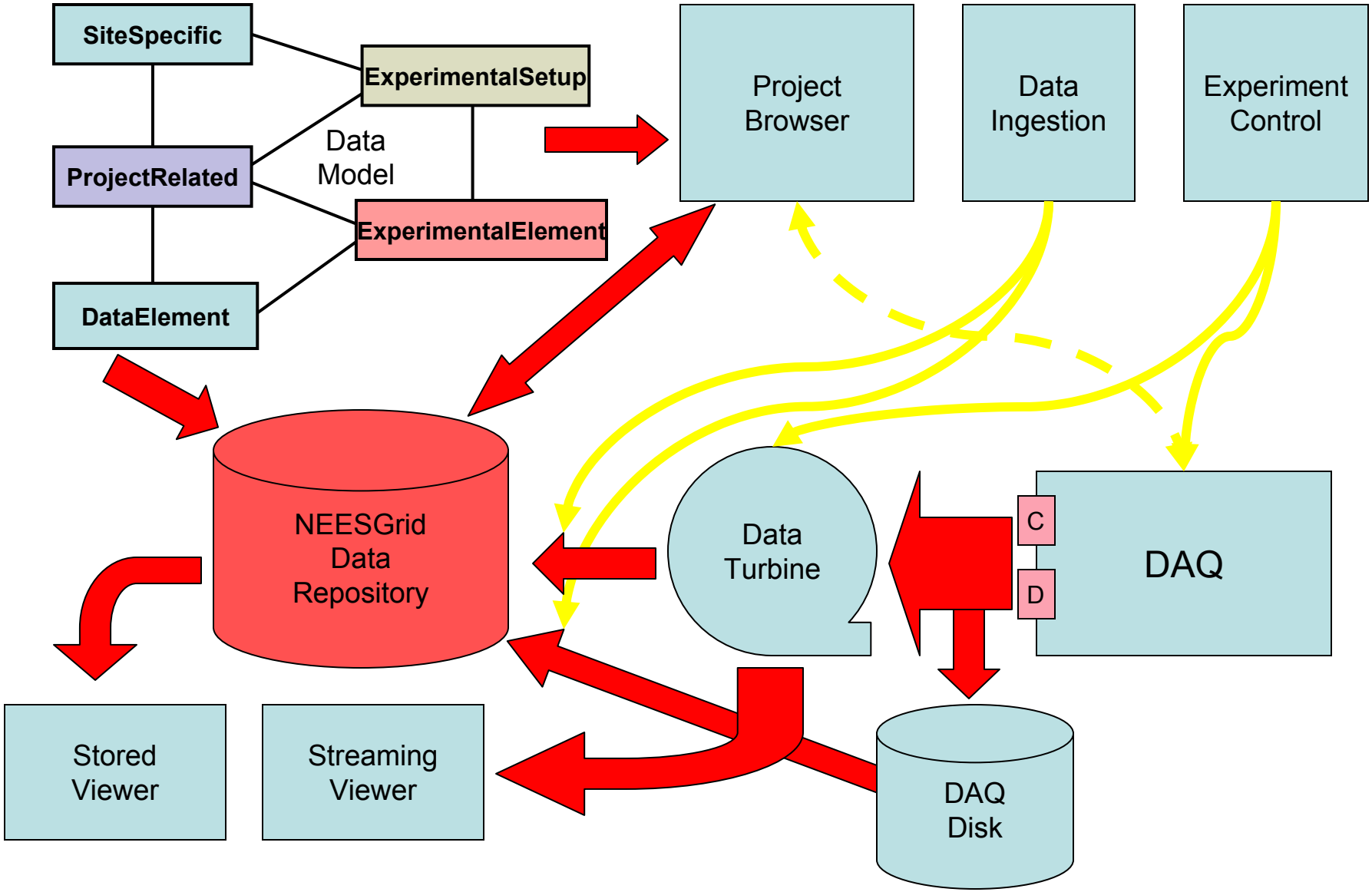

Data
RDF
Load
CHEF (dev-local): NEES: NEES: Metadata Browser - Mozilla {Build ID: 2003052908}
NEESgrid
Building the National Virtual Collaboratory for Earthquake Engineering.
Logout
My Workspace
NEES : Metadata Browser
Meta Browser
Browse By:
nees:Project
Go
Expand All
Collapse All
New $currenttype
nees:File
nees:Time
nees:Records
nees:Movie
nees:NumericalFile
nees:Unit
nees:SensorGroupReading
nees:Location
nees:Descriptive
nees:CalibrationSet
nees:SpatialLocation
nees:Book
nees:Project
nees:Visual
nees:TechnicalReport
nees:Company
nees:Proceedings
nees:Specimen
nees:Committee
nees:Publications
Home
Chat
Dataviewer
Metadata Browser
Schedule
Resources
Chat
Discussion
Dataviewer
Repository
GridFTP
Grid Cert Info
Video Cameras
Testbed Status
E-Mail
Done
CHEF (dev-local): NEES: NEES: Project Browser Test - Mozilla {Build ID: 2003052908}
Chef
CompreHensive collaborativE Framework
My Workspace
NEES
NEES : Project Browser Test
NEES:Project Browser
Home
Chat
Project Browser Test
Schedule
Resources
Chat
Discussion
Dataviewer
Repository
GridFTP
Grid Cert Info
Video Cameras
Testbed Status
E-Mail
Entry Type
Experiment
Title
Shake Table tests of a Simplified Two-Story Single-Fa
Project
CUREe-Caltech Woodframe Project
Facility
Short Description
Shake table test descriptions, test phase descriptions, instrumentation, and construction drawings
Long Description
One clear issue that emerged from the Invitational Workshop on Seismic Testing, Analysis and Design of Woodframe Construction, organized by Element 1 of the CUREe-Caltech Woodframe project, was the lack of understanding of the seismic behavior of woodframe systems. Very few numerical models capable of analyzing the seismic behavior of
Done

# Experiment Preparation

- Notebook
  - Allows the creation of *material* without needing a model
  - The model is pages, chapters, and “stuff”
  - It is all captured with data and metadata
  - A notebook can be attached to any object in the model structure (i.e. a project can have a notebook, a trial can have a notebook, etc…)
- Resources
- Discussions
- Project Browser
  - Setup basic structured metadata for the experiment - Trials, descriptions, sensors, etc… This material is captured in accordance to and with the data model


CHEF (dev): NEES: NEES: E-Notebook
http://neespop.si.umich.edu:8086/chef/portal/group/nees/js_
Google
NEESgrid
Building the National Virtual Collaboratory for Earthquake Engineering.
Logout
My Workspace
NEES
Home
Chat
Schedule
Resources
Chat
Discussion
Dataviewer
Repository
GridFTP
Grid Cert Info
Video Cameras
Testbed Status
E-Mail
E-Notebook
Applet ELNConnectionAp
Electronic Notebook ELN_5.1 - http://neespop.si.umich.edu:8086/sam
File
Edit
Search
Admin
Help
test42
fgbhnjmkl
fghjbknlm
Chuck's Fun Stuff
Status
Add a note to: Chuck's Fun Stuff
Editors
Image snap
Load file
Whiteboard
URL Editor
Equation
Text
HTML text
Add Entry
Name:
Description:
Note Addition:
Displayed:
Submit
Clear All
Clear Item
Edit

# NEESgrid Experiment Data Flow

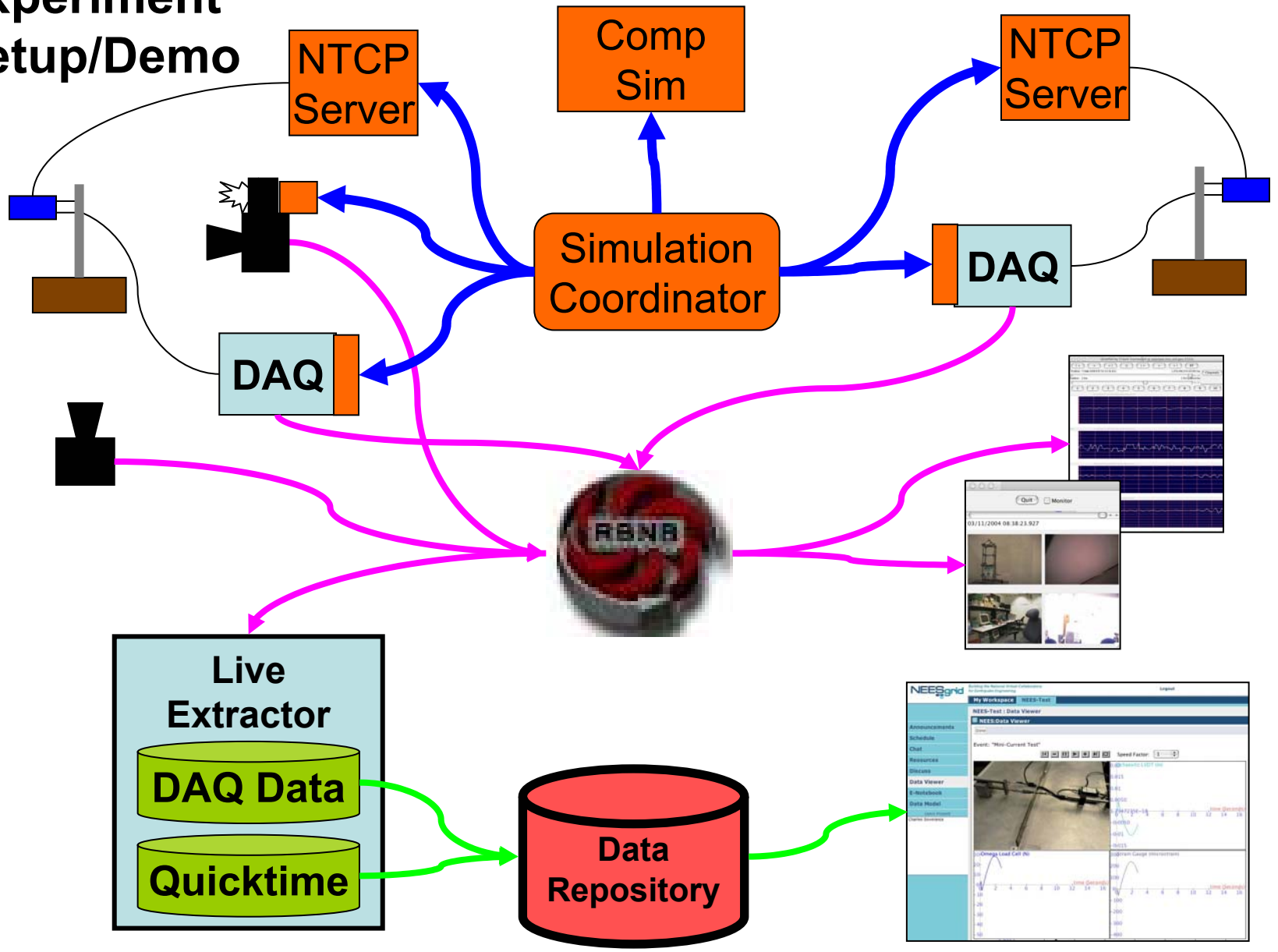


# Data Turbine

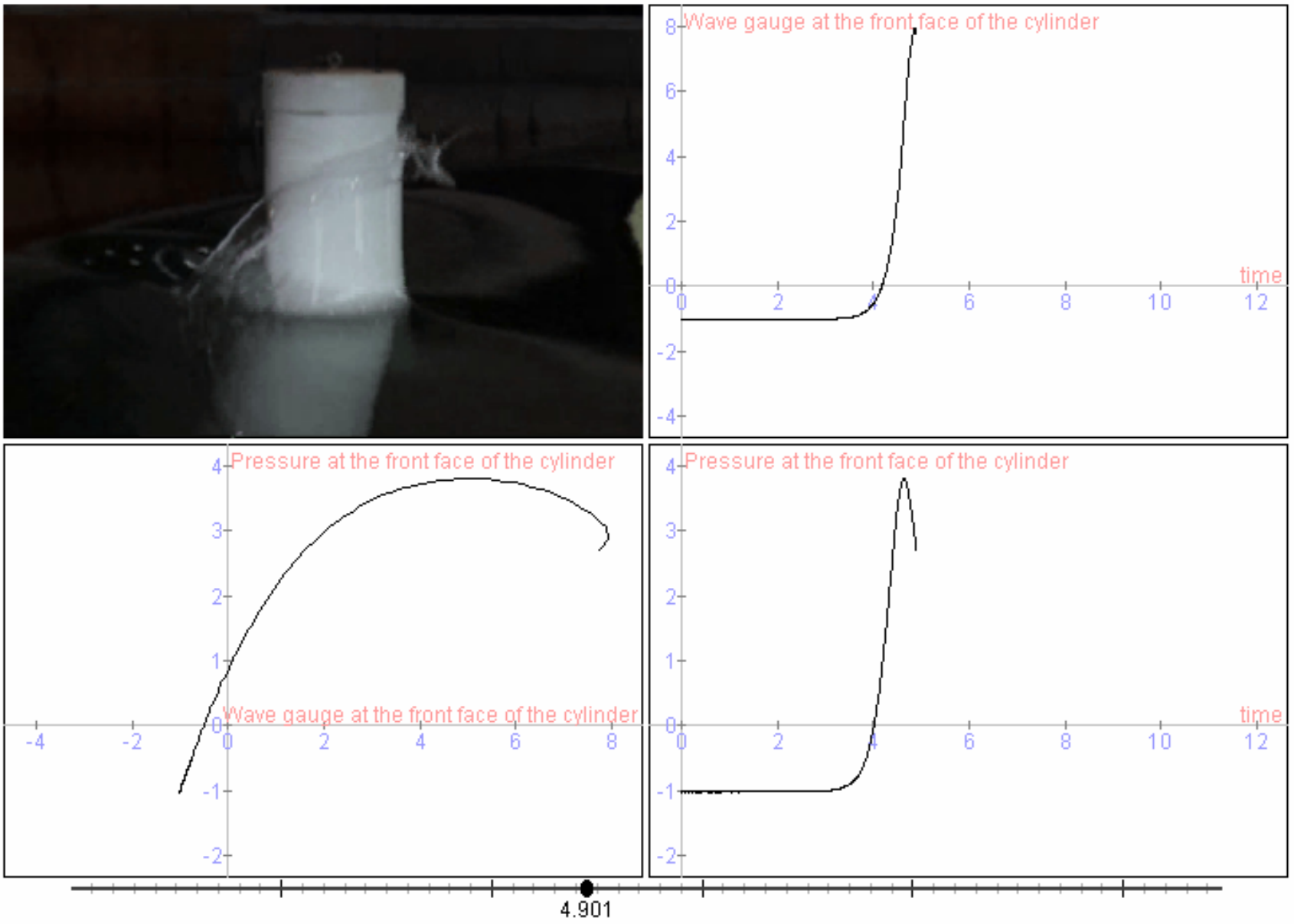
- Dynamic data server that provides a unified view of static and streaming data for universal data access
  - Video and multimedia
  - Test data acquisition
  - Telemetry streams
  - Real time monitoring
  - Delay tolerant networking


- Highly scalable by allowing linkage of multiple data turbine servers
- Interfaces to Matlab and Labview

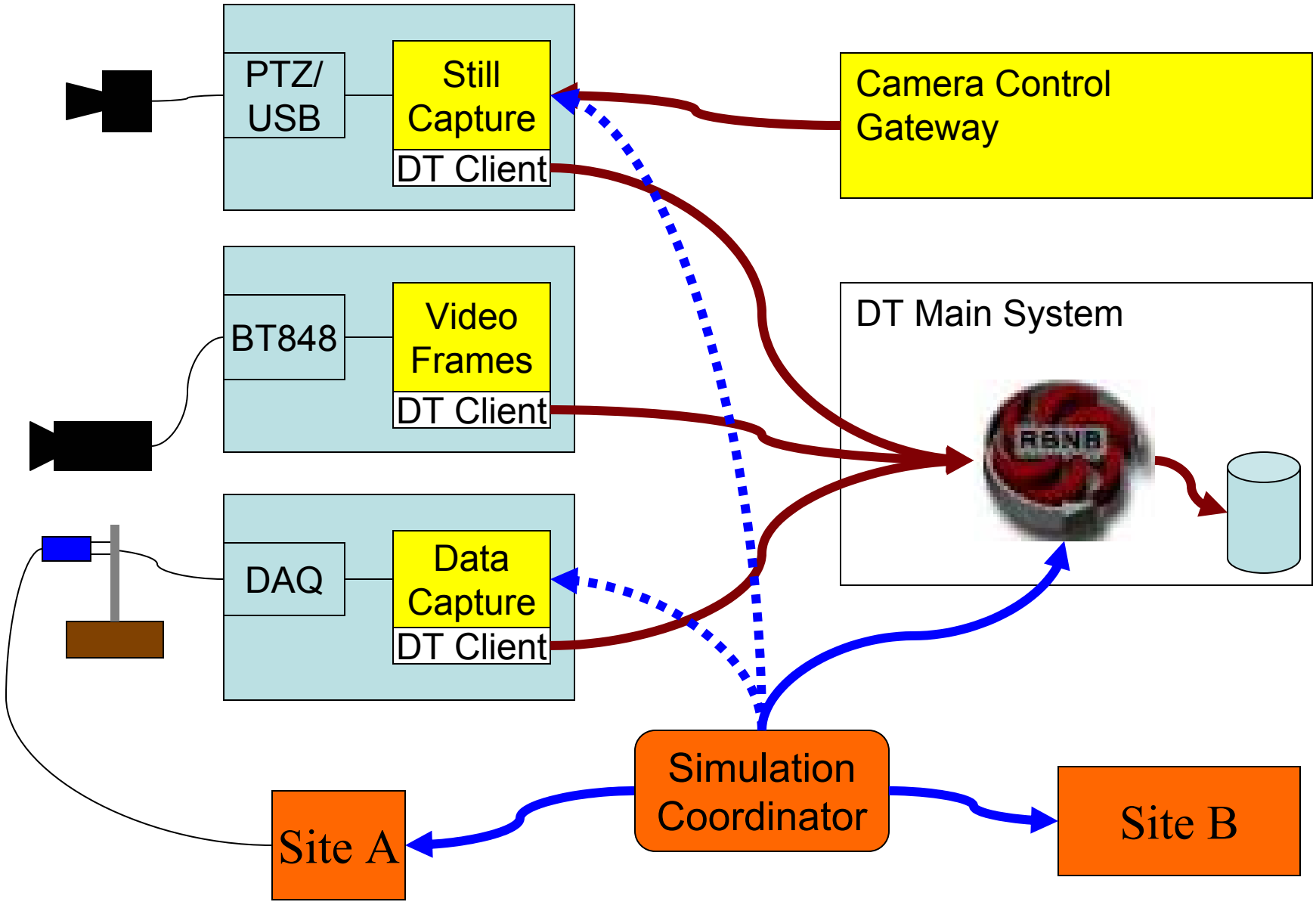
# Experiment Setup/Demo



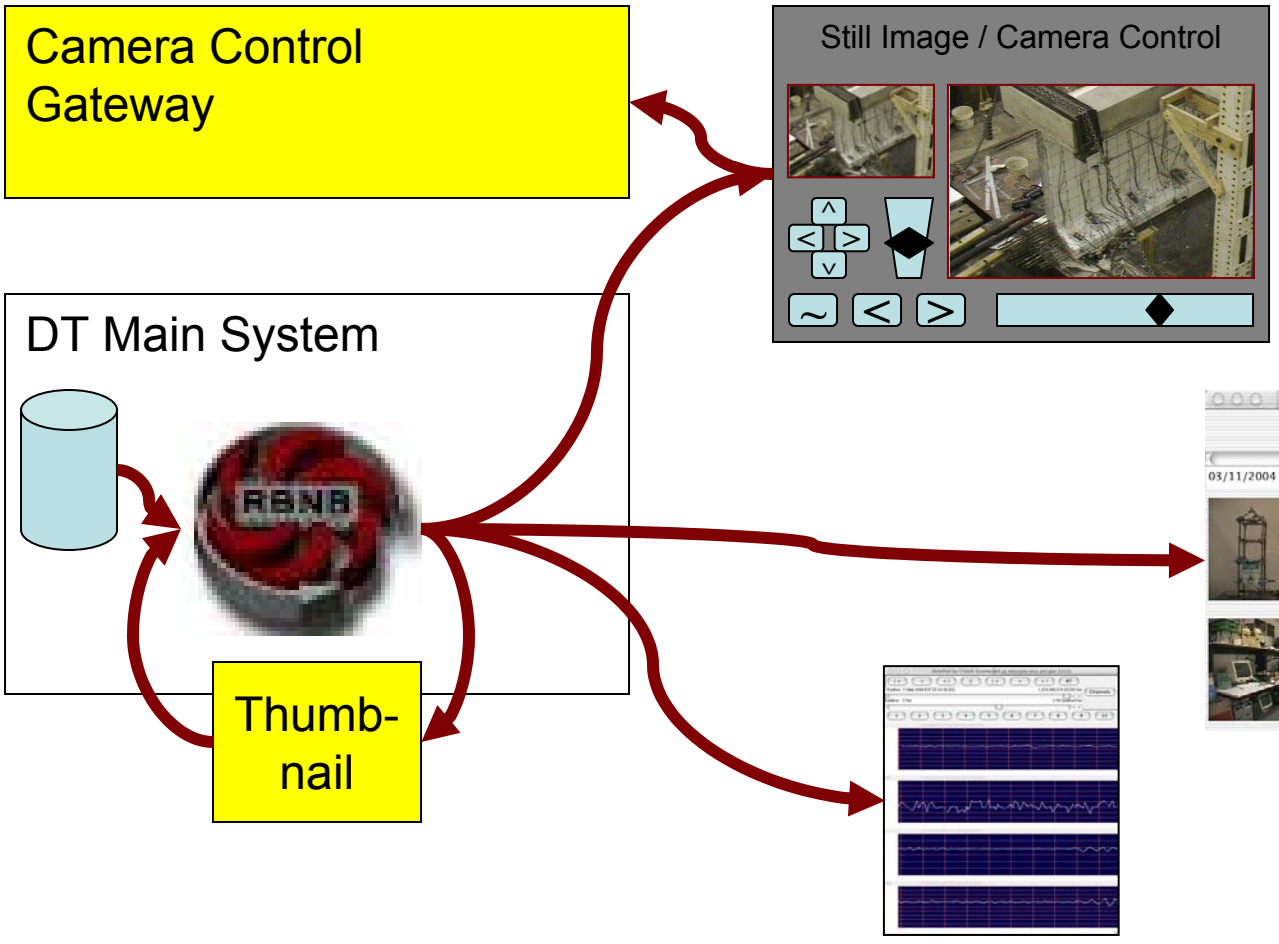
Event: "Oregon Large Tank Test September 8, 2003"



# Capturing Video and Data



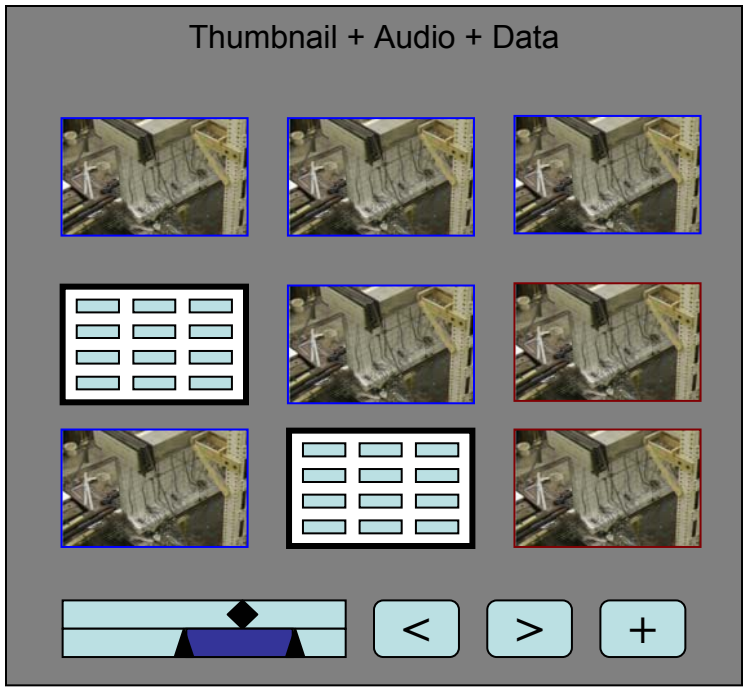
# Data Monitoring Tools




Still image camera control

Creare viewers

# Video and Data Tivo




# Summary

- As a Grid portal such as NEESGrid is developed, it reveals many requirements that were only vaguely understood before software development started.
- As “things” without user interfaces gain user interfaces, hidden flaws in the underlying “things” are revealed and must be fixed…
- The portal effort is not just a technical job, it becomes one of the major transformative catalysts for the field.
- Be careful assuming that you “know” too much at the beginning of the project.

# Overall Summary

- There are many design choices and opportunities when developing a Grid Portal.
- JSR-168 and WSRP have turned to the Portal world upside down and given a chance to re-think many aspects of portals.
- While there is much complexity, the first task is to focus on using JSR-168 to build a set of basic reusable portlets to do the rather generic jobs.
- The Sakai effort is best though of as many portlets written for a particular task rather than a portal technology itself.
- When Sakai is completed it can be blended together with other JSR-168 portlets to produce a collaborative Grid Portal.

# Questions

- Thank you for your time.
- On to the JSR-168 tutorial…