Web Services for Visualization

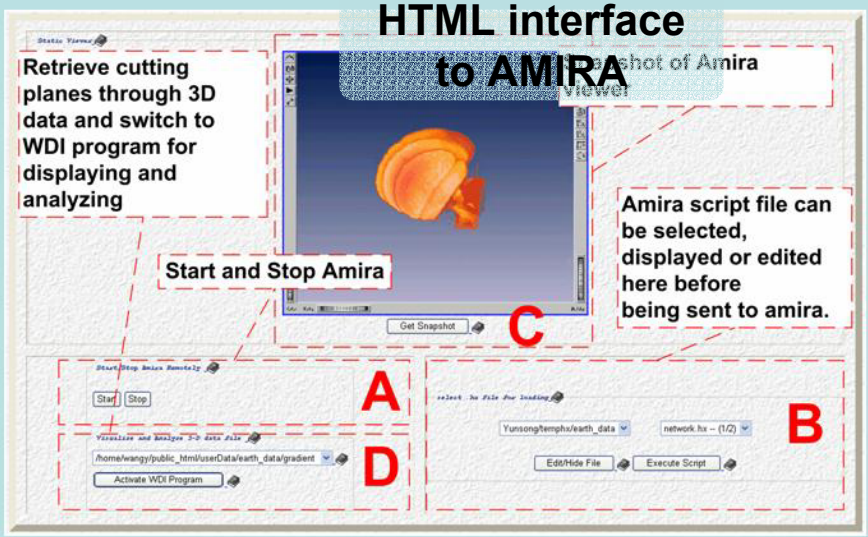
Gordon Erlebacher (Florida State University)

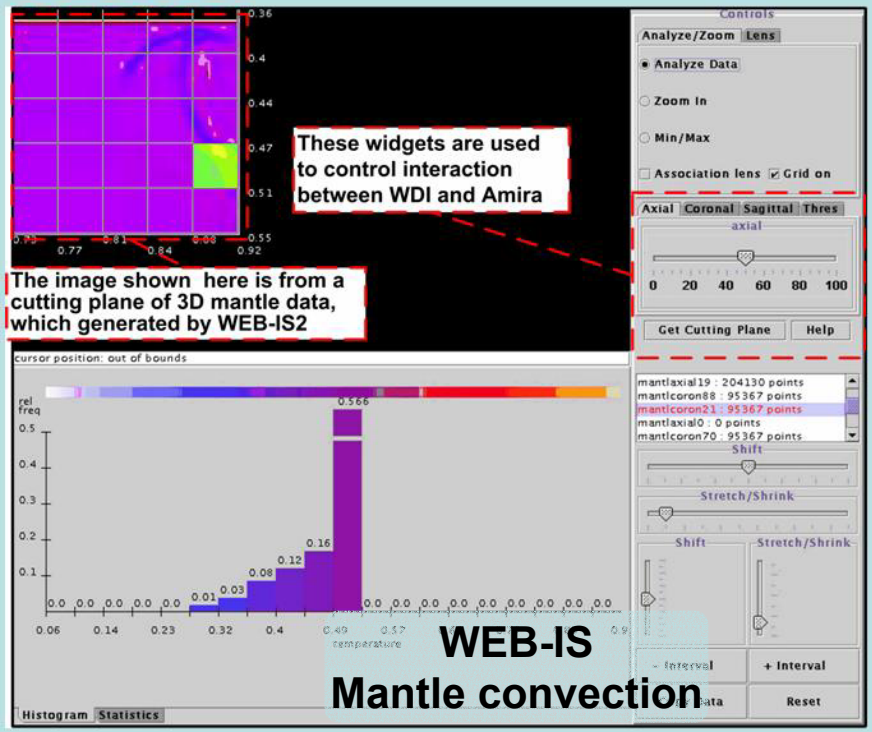
Collaborators:
S. Pallickara, G. Fox (Indiana U.)
Dave Yuen (U. Minnesota)

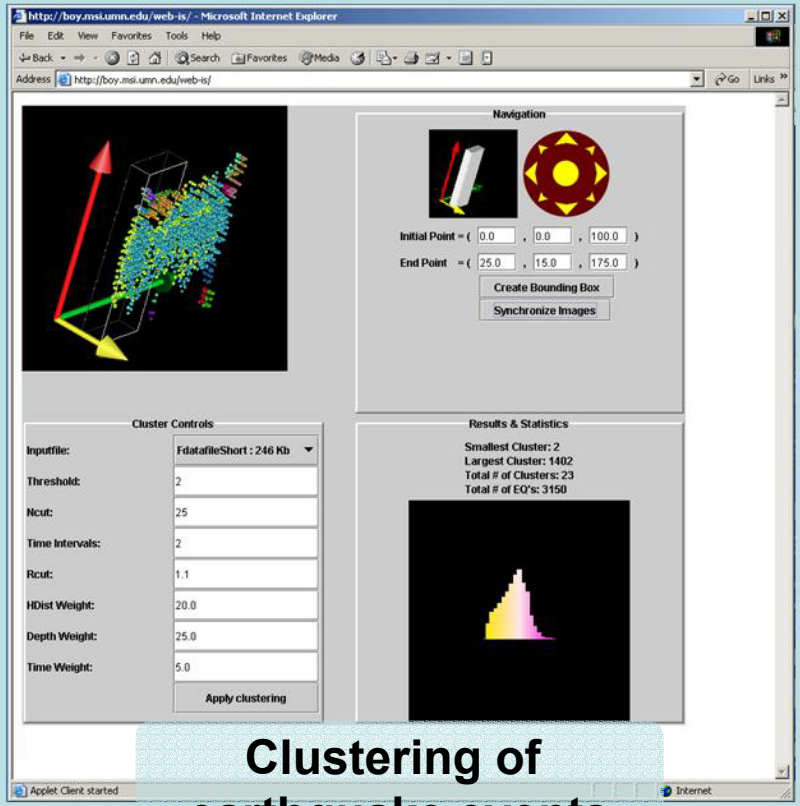
State of affairs

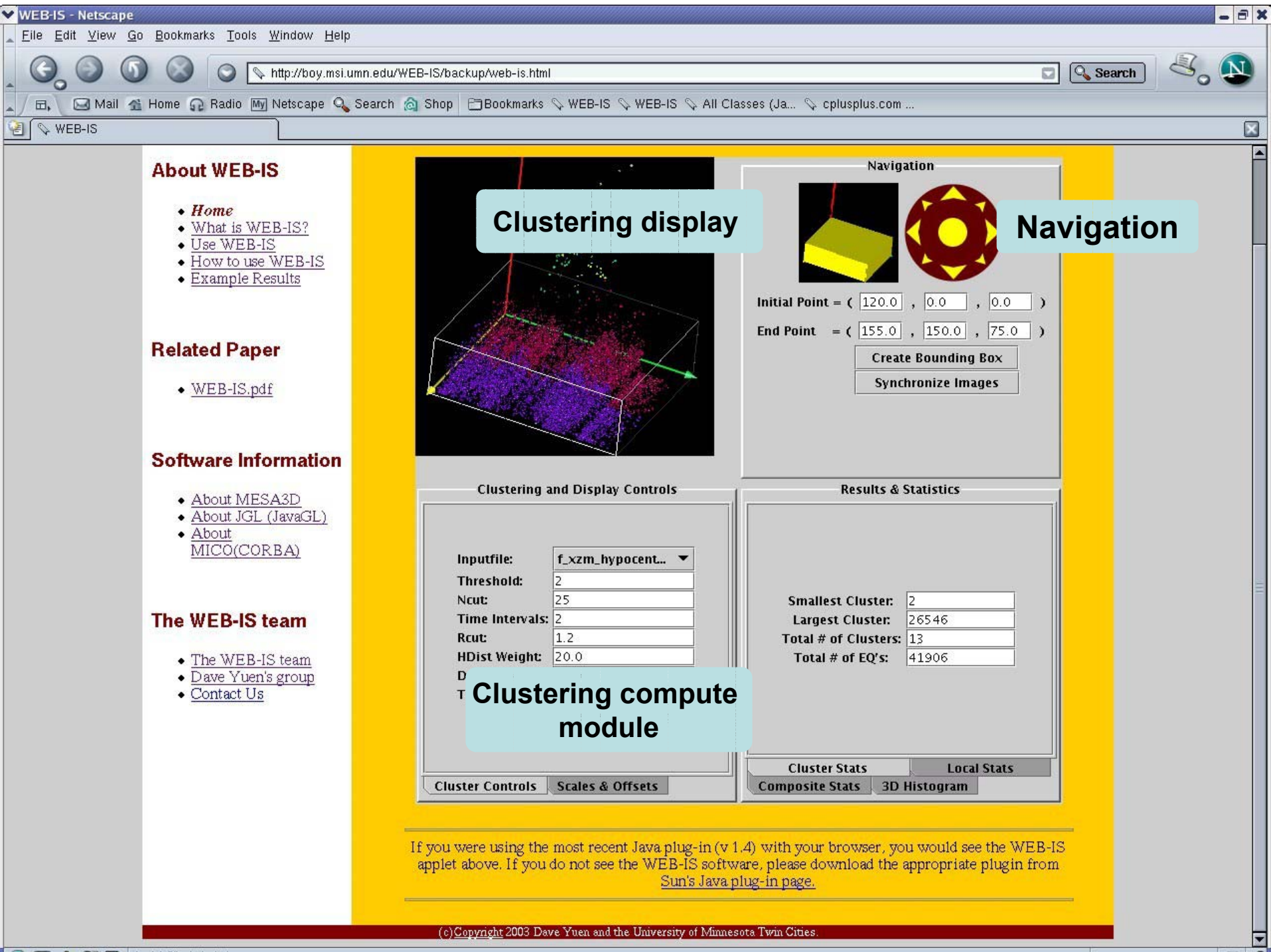
- Size of datasets is growing exponentially fast
- Client devices are proliferating with wide range of functionality (PDA, laptop, phones, desktop, powerwalls, etc.)
- Bandwidth is increasing (Internet 3: 10 Gbit/sec), but insufficiently fast
- Increasing collaborations between geographically distributed scientists
- Complexity of systems makes them increasingly unreliable
- Clusters and Grids are becoming ubiquitous
- Promised simplicity for the user is often not delivered

Clients


HTML interface to AMIRA
Retrieve cutting planes through 3D data and switch to WDI program for displaying and analyzing
Start and Stop Amira
Amira script file can be selected, displayed or edited here before being sent to amira.
Get Snapshot
A
B
C
D
Start
Stop
Activate WDI Program
Edit/Hide File
Execute Script


These widgets are used to control interaction between WDI and Amira
The image shown here is from a cutting plane of 3D mantle data, which generated by WEB-IS2
Controls
Analyze/Zoom
Lens
Analyze Data
Zoom In
Min/Max
Association lens
Grid on
Axial
Coronal
Sagittal
Thres
Get Cutting Plane
Help
Shift
Stretch/Shrink
Histogram
Statistics
WEB-IS
Mantle convection

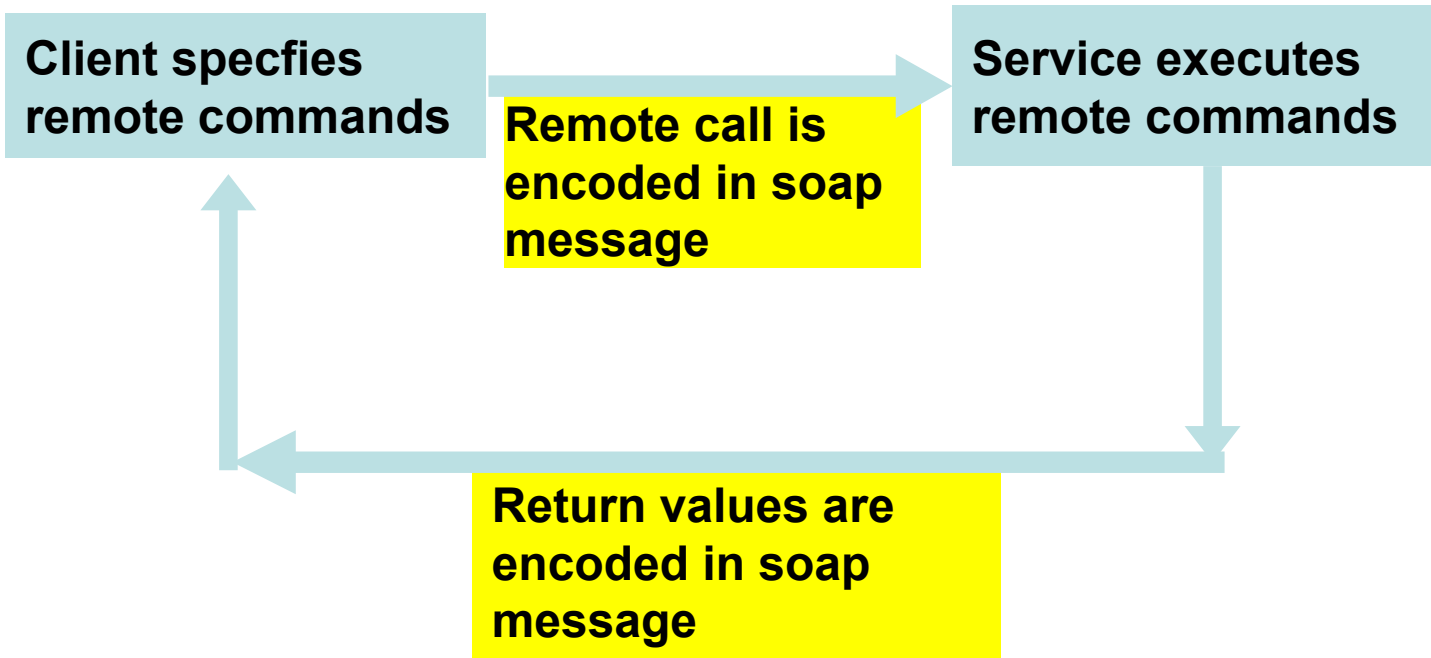

Navigation
Initial Point = (0.0 , 0.0 , 100.0)
End Point = (25.0 , 15.0 , 175.0)
Create Bounding Box
Synchronize Images
Cluster Controls
Inputfile:
FdatafileShort : 246 Kb
Threshold:
Ncut:
Time Intervals:
Rcut:
HDist Weight:
Depth Weight:
Time Weight:
Apply clustering
Results & Statistics
Smallest Cluster: 2
Largest Cluster: 1402
Total # of Clusters: 23
Total # of EQ's: 3150
Clustering of earthquake events


WEB-IS - Netscape
File Edit View Go Bookmarks Tools Window Help
http://boy.msi.umn.edu/WEB-IS/backup/web-is.html
Search
Mail Home Radio Netscape Search Shop Bookmarks WEB-IS WEB-IS All Classes (Ja... cplusplus.com ...
WEB-IS
About WEB-IS
Home
What is WEB-IS?
Use WEB-IS
How to use WEB-IS
Example Results
Related Paper
WEB-IS.pdf
Software Information
About MESA3D
About JGL (JavaGL)
About MICO(CORBA)
The WEB-IS team
The WEB-IS team
Dave Yuen's group
Contact Us
Clustering display
Navigation
Navigation
Initial Point = (120.0 , 0.0 , 0.0)
End Point = (155.0 , 150.0 , 75.0)
Create Bounding Box
Synchronize Images
Clustering and Display Controls
Inputfile: f_xzm_hypocent...
Threshold: 2
Ncut: 25
Time Intervals: 2
Rcut: 1.2
HDist Weight: 20.0
Clustering compute module
Cluster Controls
Scales & Offsets
Results & Statistics
Smallest Cluster: 2
Largest Cluster: 26546
Total # of Clusters: 13
Total # of EQ's: 41906
Cluster Stats
Local Stats
Composite Stats
3D Histogram
If you were using the most recent Java plug-in (v 1.4) with your browser, you would see the WEB-IS applet above. If you do not see the WEB-IS software, please download the appropriate plugin from Sun's Java plug-in page.
(c)Copyright 2003 Dave Yuen and the University of Minnesota Twin Cities.

Partial Wish List

- Redundancy of visualization, computational, storage services
- Support for data files stored redundantly
- No single point of failure (fault tolerance)
- Automatic resource discovery transparent to the users
- Decoupling of the development of client interfaces and resource services
- User interfaces are all built into the leading browsers: Internet Explorer, Konqueror, Mozilla
- Accommodate users without the latest java plug-in technology
- Scalability of the middleware component
- Clients and services can be written in any language

Message Flow with SOAP



Services

- A service is simply a task
- Instead of execution at local machine, a service may be executed remotely
- Generally: services are accessed via point-to-point access (URL/IP of end point is known)
- Service discovery: seek service in a database, which returns the URL/IP
- More generally: request access to service by name, and have system choose one for you

Services: examples

- **Visualization and graphic services**
  - Image processing
  - Scientific visualization
  - Video streaming
  - Creating of videos
- **Computation services**
  - Statistical calculations
  - Symbolic manipulation
  - Data filtering and feature extraction
  - Data transforms (e.g., wavelet)
- **Storage services**
  - File retrieval, subset selection
  - File caching

Solution: Publish/Subscribe

Prototypical Example:

a news Server

News Server: publishers

- Topics are the newsgroups labels, e.g.,
  - Topic: PC hardware
  - Topic: PC software
  - Topic: Mac hardware
  - Topic: OpenGL developers
- Anyone can publish to any topic
  - Subject to administrative restrictions
  - Can publish at any time

News Server: subscribers

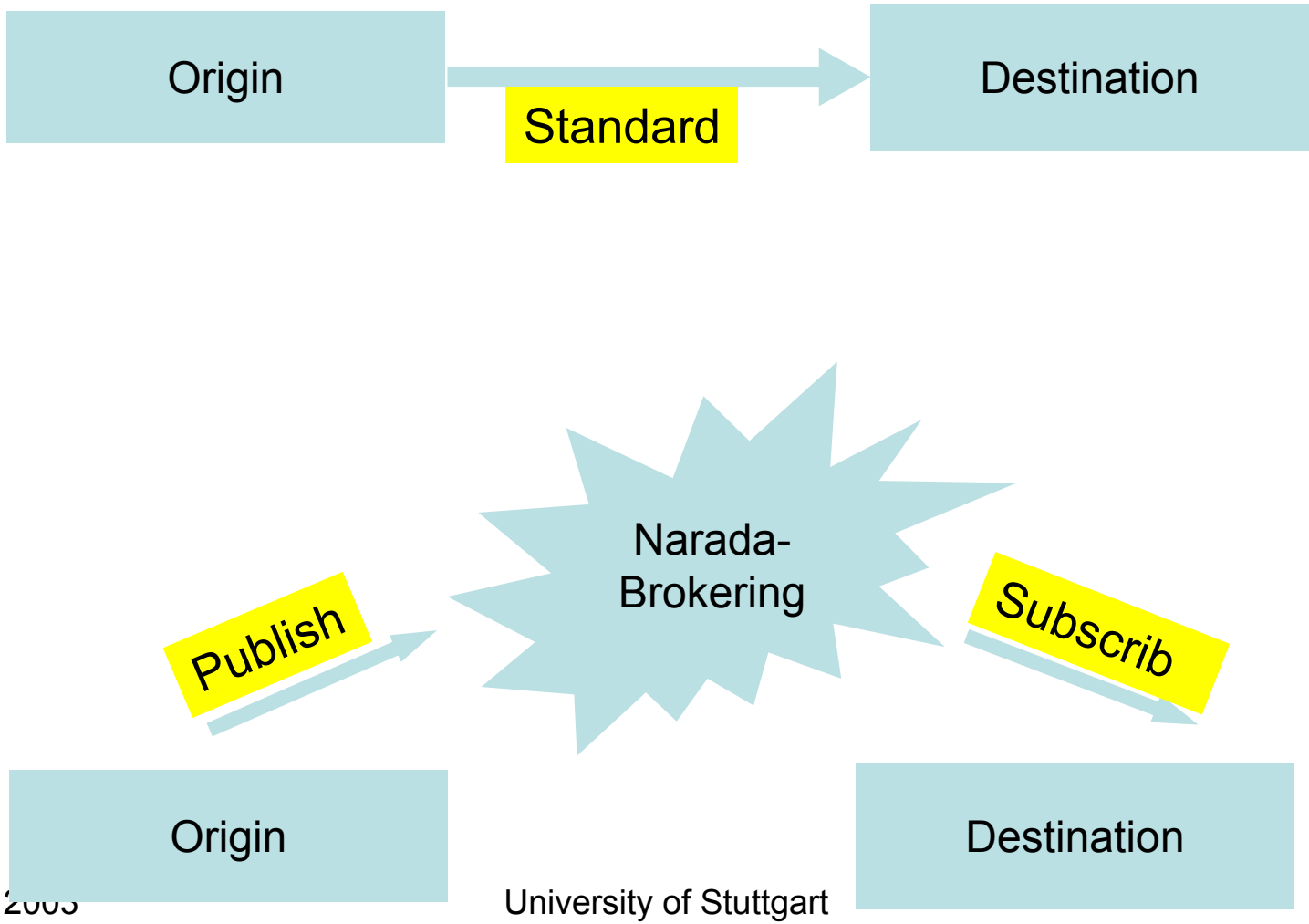
- Users subscribe to particular news groups
- Updating is equivalent to retrieving all news items from all groups subscribed to
- As new items are published (i.e., posted) to the groups, they can be forwarded to all the subscribers
- User can subscribe to a non-existent news topic. If (and when) the topic is created, the subscriber will receive the news item.

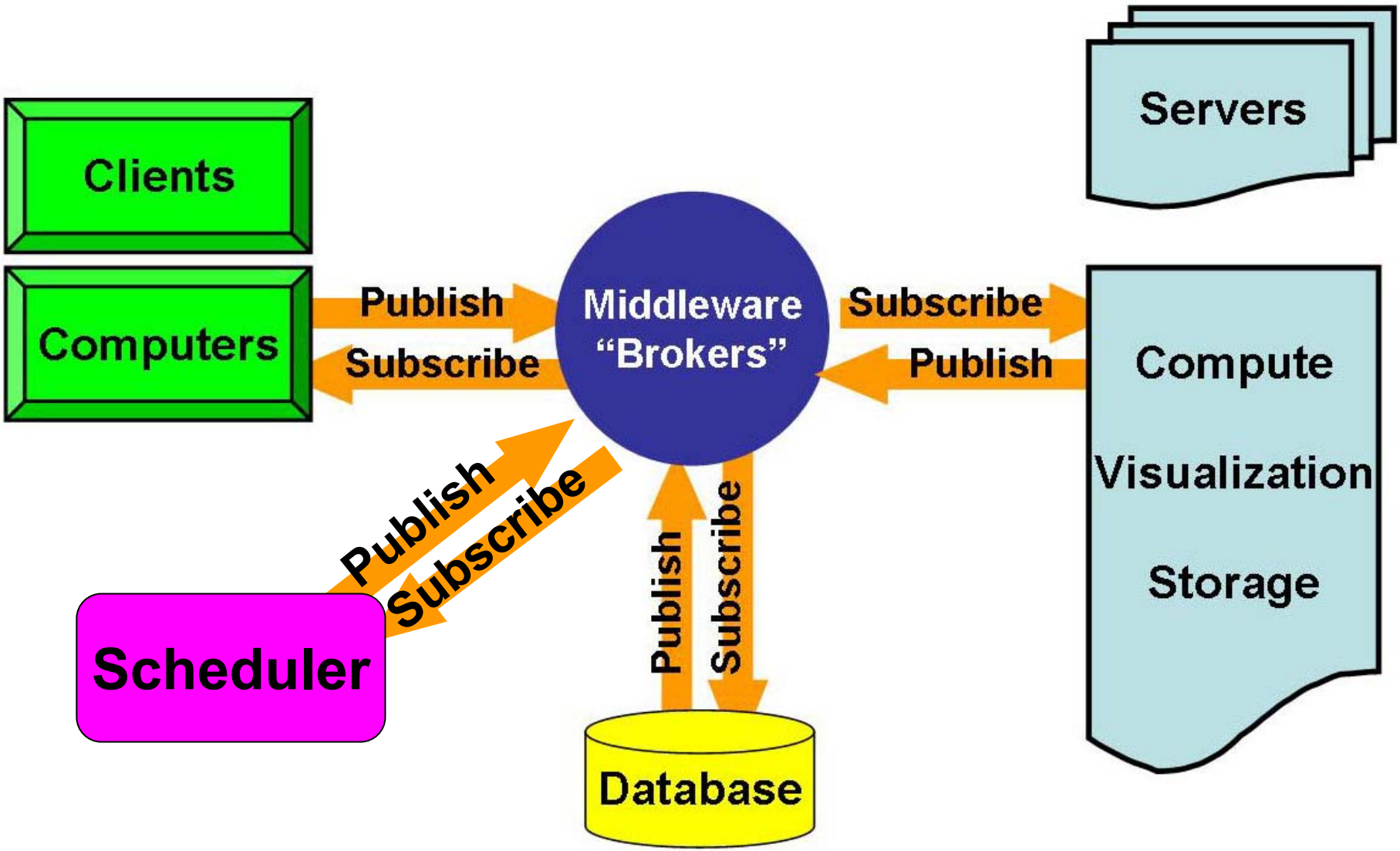
NaradaBrokering (NB)

Developer: S. Pallickara, Indiana Univ.

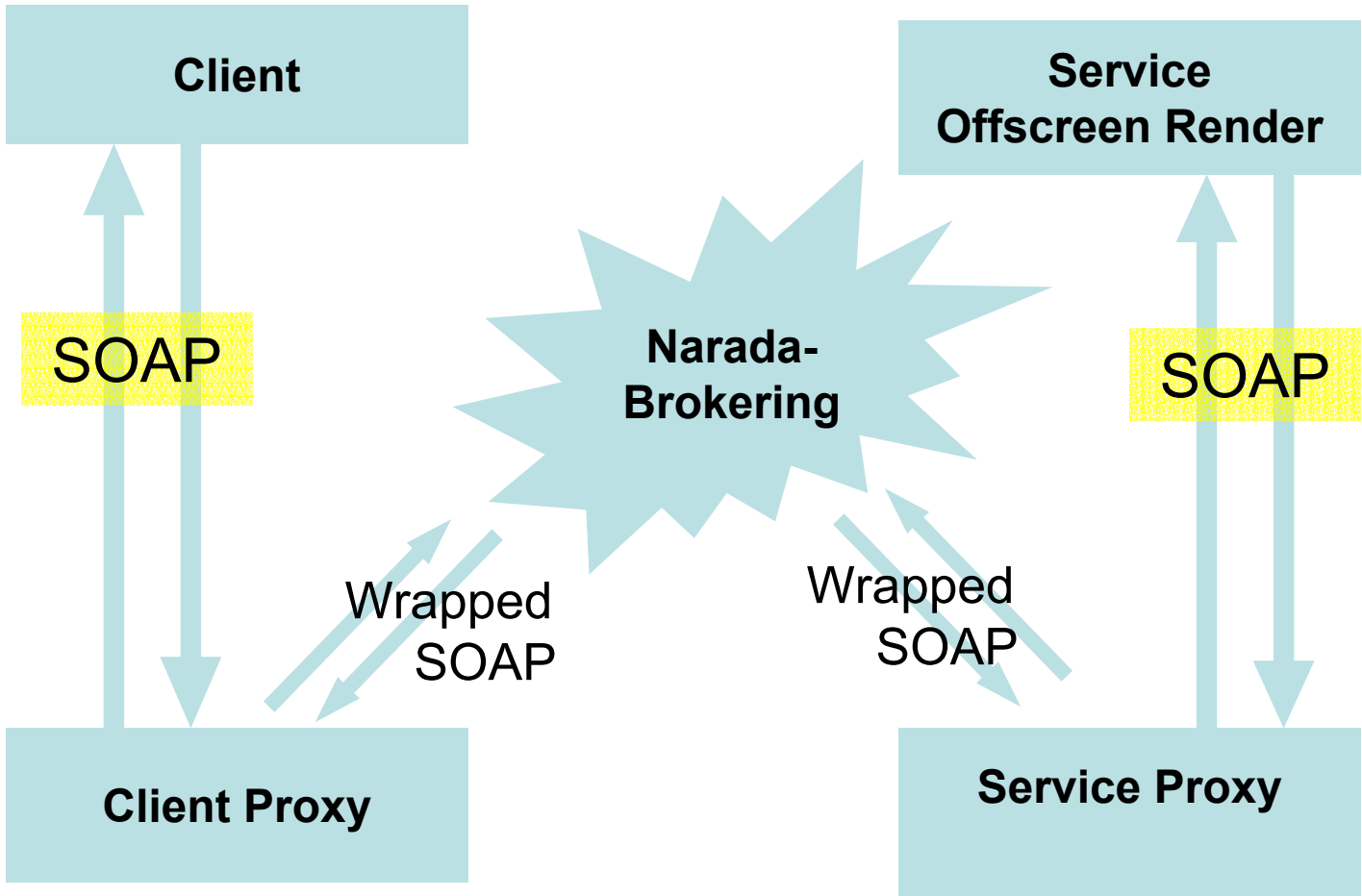
- Distributed Messaging Infrastructure
- Messages are tagged by a *topic*
- Multiple publishers send messages to the same topic
- Multiple subscribers receive messages from a given topic
- Any client/service can subscribe or publish to multiple topics
- Messages have a lifetime
  - Transient: they disappear once any subscriber consumes it
  - Indefinite: message stays in the system until explicitly removed by some outside action
- Network of “brokers”
- Nodes can be added and removed from the network without affecting functionality

Message Flow




Servers
Clients
Computers
Publish
Subscribe
Middleware
"Brokers"
Subscribe
Publish
Compute
Visualization
Storage
Publish
Subscribe
Scheduler
Publish
Subscribe
Database

Messaging with SOAP and NB



Proxies

- Soap messages become body of a NB message
- Messages identified by topic
- Only proxies can publish and subscribe to NB
- Proxies shield clients and servers from any knowledge of NB

Advantages of scheme

- Asynchronous discovery of task schedulers, available resources, resource metadata, file locations
- Both task schedulers and resources accept tasks based on load
- Ability to maintain audit trails to track system usage, task execution, and task execution failures
- In the event of failures, return of partial results and the status of the computation
- Built-in collaboration mechanisms: task results can be viewed by multiple users sharing one or more task IDs.
- Task updates are published to a “Task update” topic to allow clients to keep track of task execution
- Ability to override previous task submissions in the light of changing conditions

Advantages (cont.)

- Ability to cache results to improve performance
- Built-in fault tolerance
  - NB is a cluster
  - Duplication of services
- Complete decoupling of clients and services through proxies
- Interrupts
  - Send message to “interrupt” topic
  - Unsuscribe topics based on message content

Disadvantages

- Increased number of intermediate “hops” between client and service:
  - Client to proxy client
  - Proxy client to NB
  - Multiple hops within NB cluster
  - NB to proxy server
  - Proxy server to server
  - … and the return path …

External Databases

- DB proxies subscribe and publish to NB on behalf of the DB
- Queries (SQL) are sent to DB in the guise of a message published to NB
- Query DB for topic information

Schedulers

- Take a high level task definition and break it into subtasks
- Submit these subtasks to appropriate services
- Perform dependency analysis on the subtasks
- Return results to the subscribers

Services

- Can be in any language which has interface to SOAP
- Executed via remote calls
- Therefore, defined in terms of an interface
- Interface is encoded into SOAP messages

Storage Services

- Retrieve files from storage
  - Supercomputer
  - File servers
  - Archival storage
- Process file (on one or multiple servers)
  - Extract subset
  - Filter subset
  - Transform subset
- Send processed file for computing or visualization services

Visualization Services

- Offscreen, remote visualization
- Hardware-enabled
  - using MesaOS (Mesa off-screen) and DRI (direct rendering interface)
- Cache partial results (perhaps a caching service)
- Wrap existing software
  - We already have developed interface to Amira (U. Berlin, Template Graphic Software)
  - Custom remote software
  - (hopefully) leverage work done at Stuttgard U. (Stegmaier and others)

Computation Services

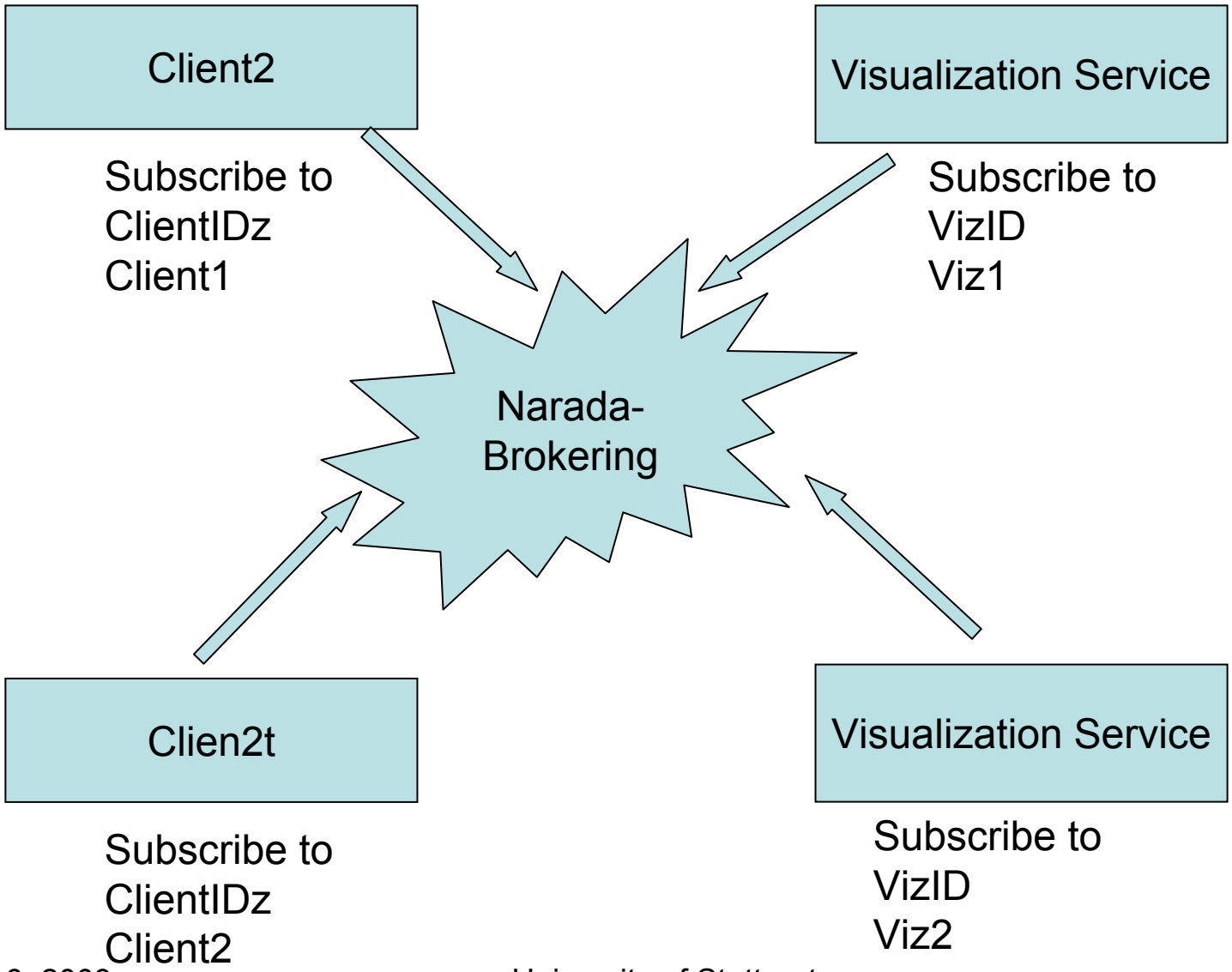
- Statistical modules
- Clustering modules
  - Earthquake clustering (10^5 – 10^6 events)
- Wavelet and filtering modules

Collaboration

- Two subscribers to the same topic can receive the same information
- Subscribers may subscribe to a visualization topic asynchronously
- Issue: how does one subscriber directly communicate with another
- How can collaborators *exchange* information

Possible bottlenecks

- When large files are transported, it may be inefficient to encapsulate them in a message to Narada.
- Images (2D) can be sent through Narada (don’t take too much bandwidth)
- Large 3D datafiles (or subsets thereof) should perhaps be transported directly to destination via point to point


Client2
Visualization Service
Subscribe to
ClientIDz
Client1
Subscribe to
VizID
Viz1
Narada-
Brokering
Clien2t
Visualization Service
Subscribe to
ClientIDz
Client2
Subscribe to
VizID
Viz2

Service discovery

- A service discovery service is responsible for providing clients with information on available services
- Can be implemented using UDDI, or some other XML-based querying language

Redundancy

- All services should be redundant. These include:
  - Databases
  - Schedulers
  - Visualization, computation, or other services
  - Service discovery
- The NB is a clustered service with built-in redundancy
- Consequence: part of NB or some services may become unavailable without impacting usage

Conclusions

- We described a new middleware fabric, which,
  - Is flexible
  - Provides loose coupling between clients and services
  - Provides fault tolerance through a publish/subscribe mechanism
  - Has the potential to develop powerful, yet flexible collaborative systems
  - Through offscreen rendering, can deliver results of visualization tools to a variety of desktops through a variety of GUIs suitable to a variety of devices
- Ease of use (and installation) will be one of our design goals!!