The iLabs Shared Architecture and the Future of Web-based Laboratory Experiments

Philip H. Bailey
MIT

mit
iCampus

Agenda

- Introduction
- WebLabs at MIT
- The iLab Shared Architecture
- The iLab Vision for the Future
- Conclusions
- Contact Information

mit iCampus

Introduction

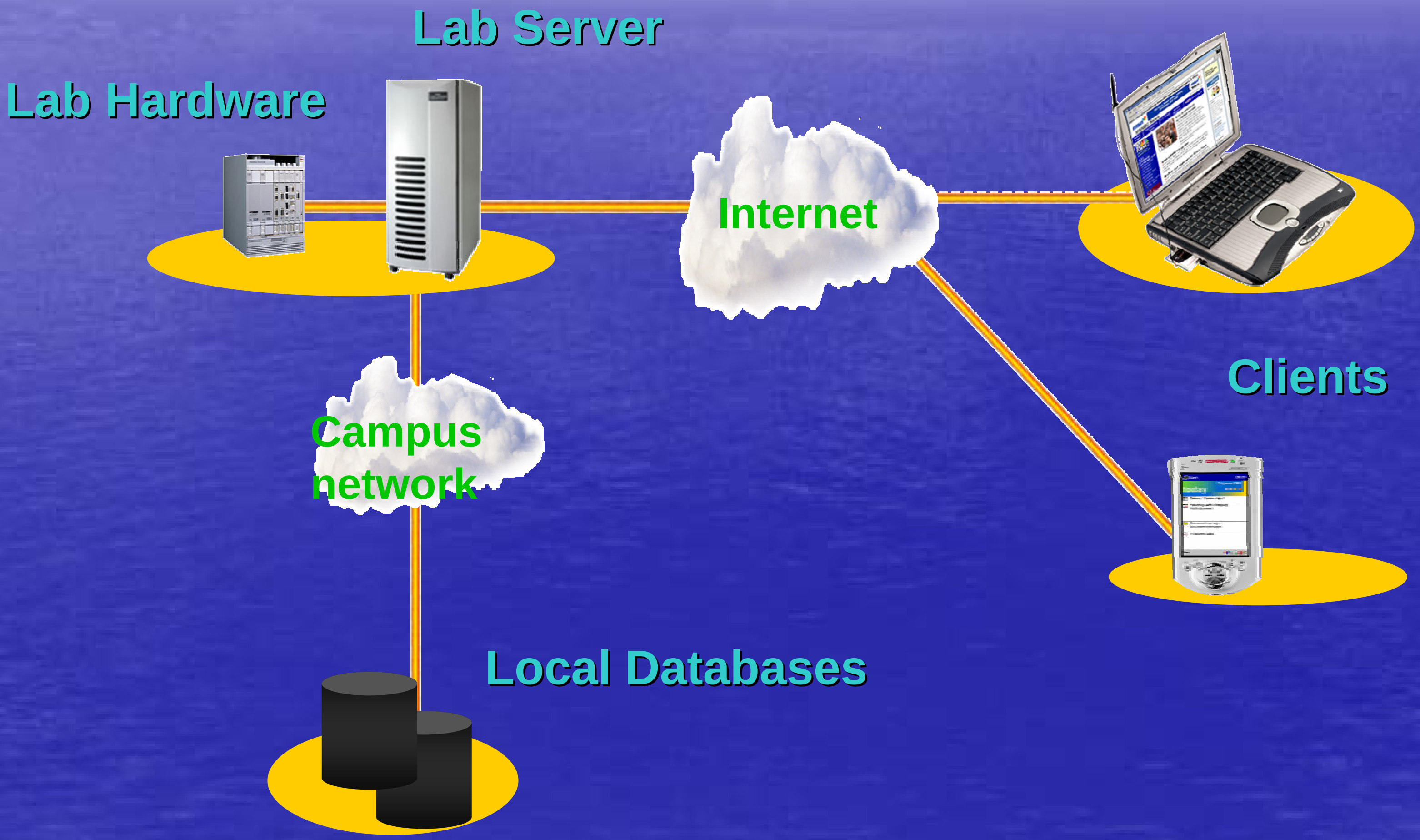
- Philip H. Bailey – MIT / Center for Educational Computing Initiatives
- Funding for the WebLab and iLab projects is provided by MIT iCampus and MIT iCampus Outreach, sponsored by Microsoft Research.

mit iCampus

WebLabs at MIT

- WebLab Overview
- MIT WebLabs
- The Student Perspective
- WebLab Benefits

Typical WebLab Architecture



Experiment Typologies

- Batched Experiments
  - The entire specification of the experiment is determined before execution begins.
  - The user need not remain online while experiment executes.
- Interactive Experiments
  - The student client controls virtual lab equipment (GUI).
  - The student can interact with experiment throughout its course.
- Sensor Experiments
  - Publish and subscribe based architecture
  - Triggers and event-driven data monitoring
  - Flexible data analysis
  - Data archive

mit iCampus

MIT WebLabs

Dynamic signal analyzer *(EECS, deployed 2004)*

Polymer crystallization *(Chem. E., deployed 2003)*

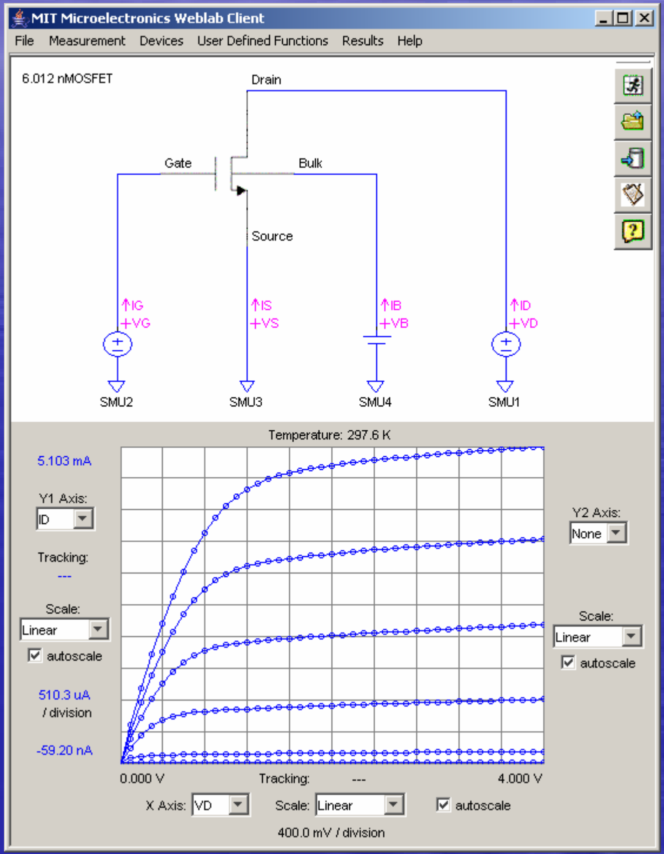
Shake table *(Civil Eng., deployed 2004)*

Heat exchanger *(Chem. E., deployed 2001)*

Microelectronics device characterization *(EECS, deployed 1998)*

Microelectronics WebLab

Goal: Characterization of Microelectronic Devices



6.012 nMOSFET

Temperature: 297.6 K

Y1 Axis: ID

Tracking: ---

Scale: Linear

☑ autoscale

510.3 uA / division

Y2 Axis: None

Scale: Linear

☑ autoscale

X Axis: VD

Tracking: ---

Scale: Linear

☑ autoscale

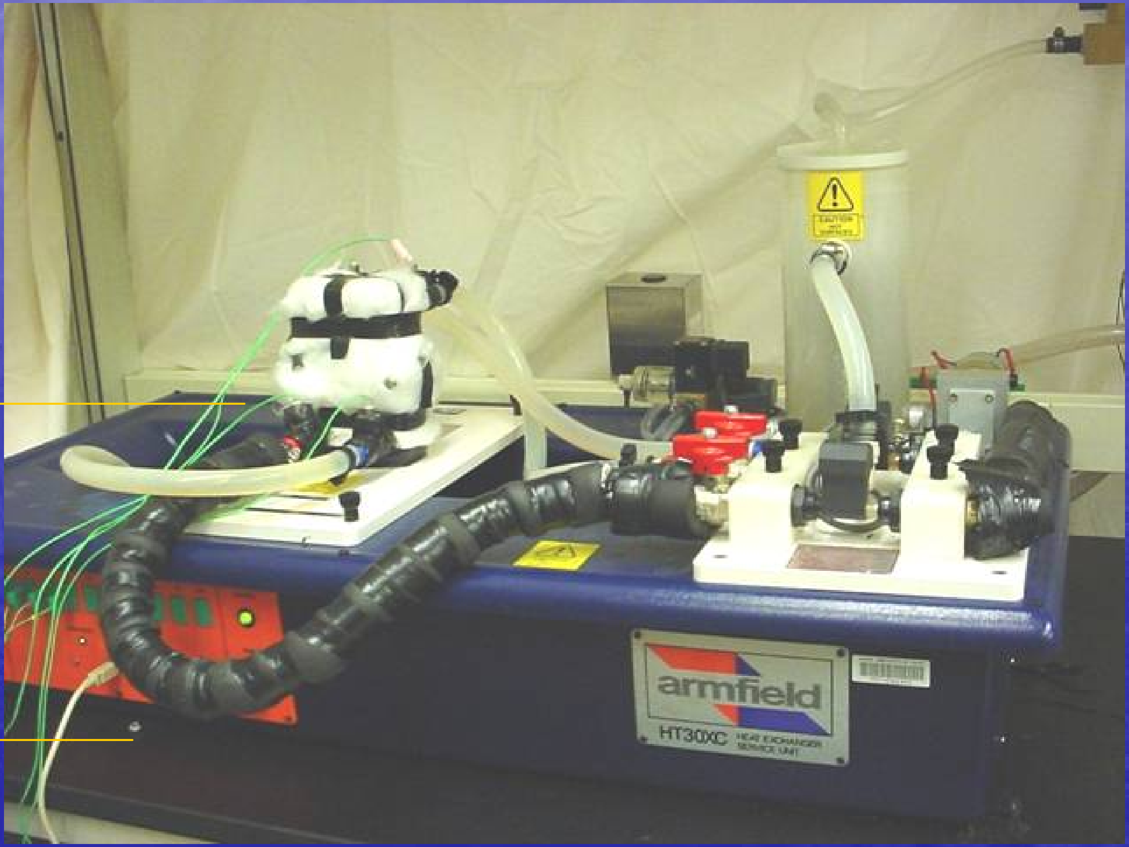
400.0 mV / division

mit iCampus

Flat Plate Heat Exchanger

Flat plate heat exchanger

Service unit



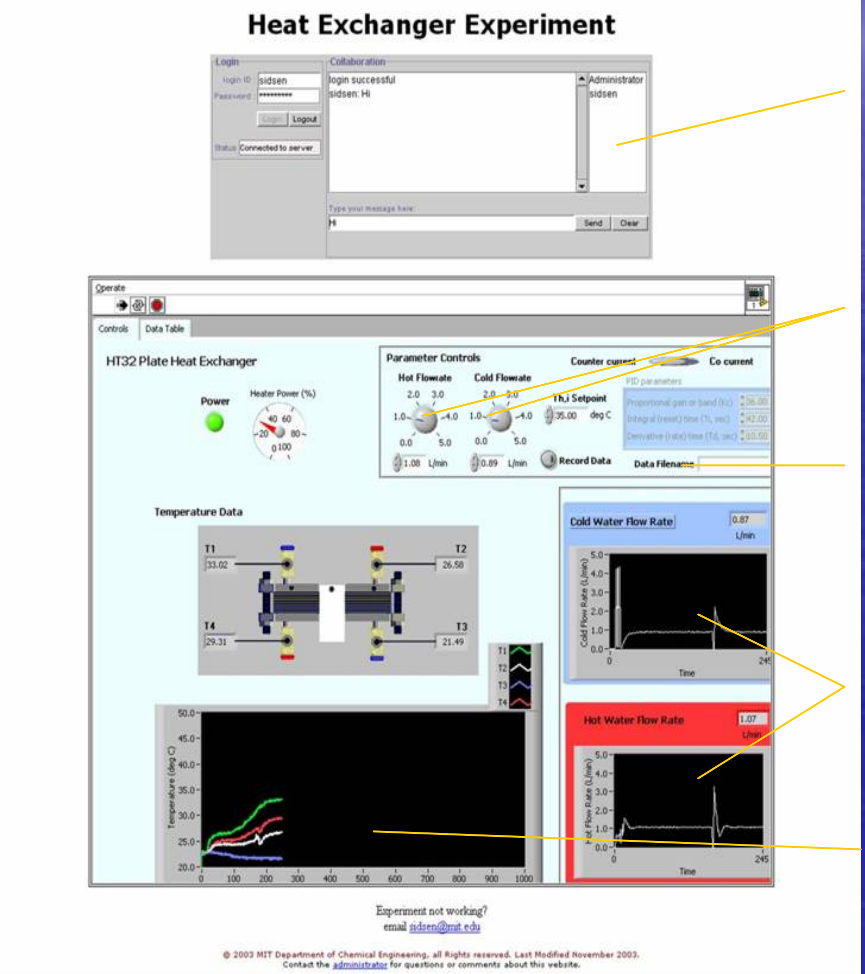
Goal:

Study dynamics of heat transfer between fluids

Heat Exchanger GUI

Java Chat Interface

LabVIEW 6.1 Interface



Heat Exchanger Experiment

Login
login ID: sidsen
Password: **********
Status: Connected to server

Collaboration
login successful
sidsen: Hi

Administrator
sidsen

Type your message here:
Hi

Experiment not working?
email sidsen@mit.edu

© 2003 MIT Department of Chemical Engineering, all Rights reserved. Last Modified November 2003.
Contact the administrator for questions or comments about this website.

List of users logged in to chat server

Hot/cold flow rate control knobs

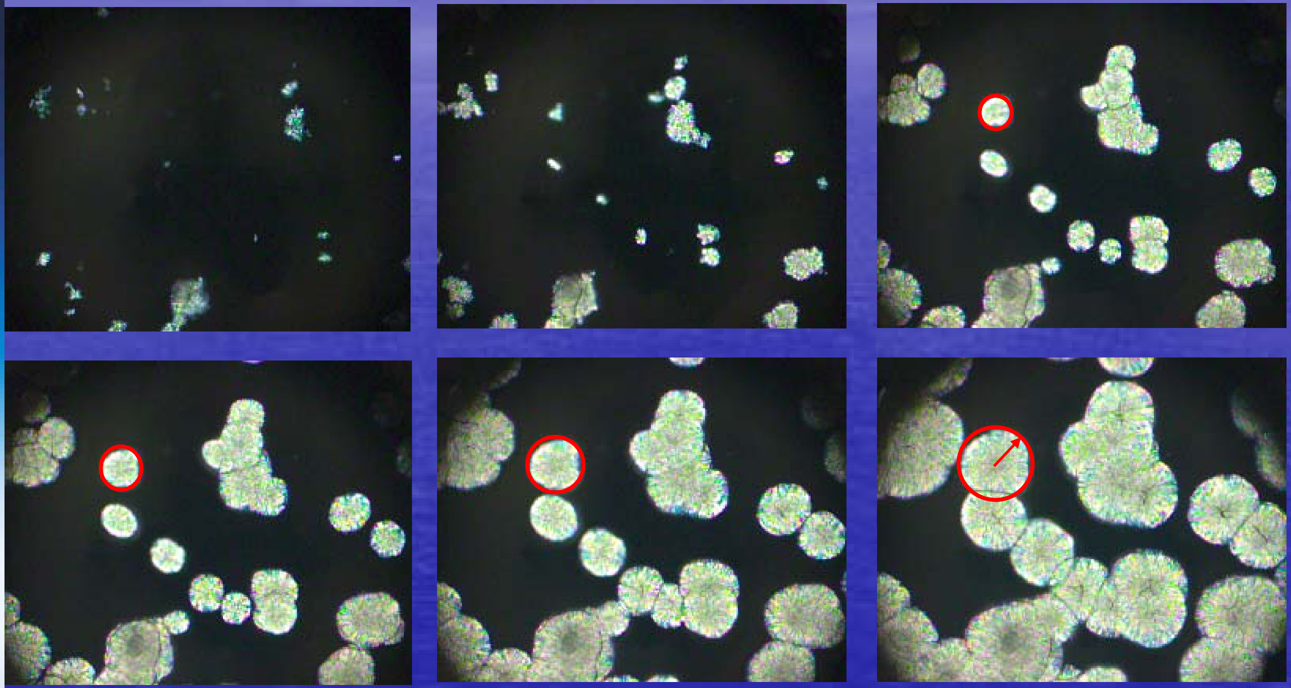
Data recording feature (exports to Excel file)

Hot/cold flow rate readings (real-time)

Thermocouple temperature readings (real-time)

mit iCampus

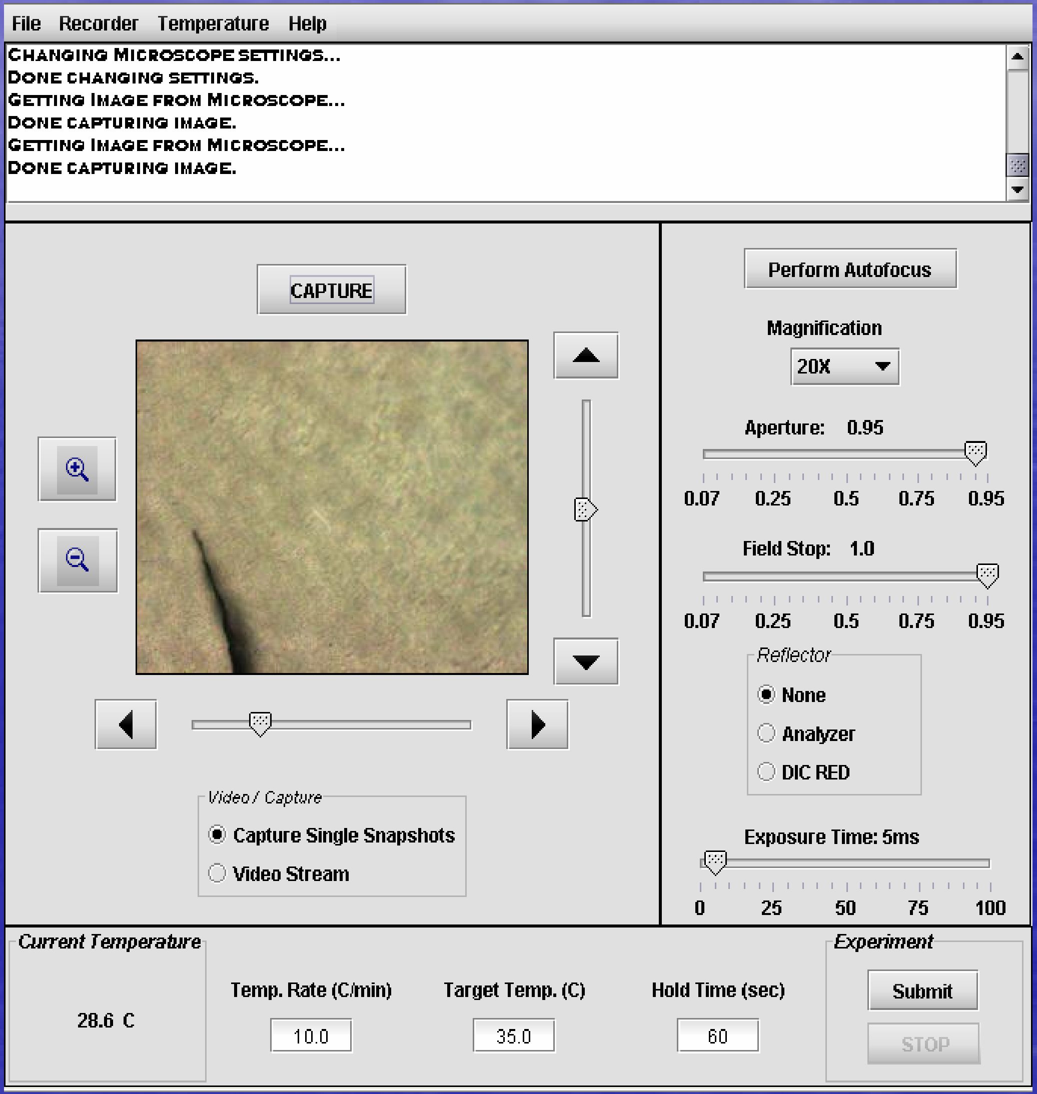
Polymer Crystallization



Goal:

Study dynamics of polymer crystallization in real time

Polymer Crystallization GUI



```
Changing Microscope settings...
Done changing settings.
Getting Image from Microscope...
Done capturing image.
Getting Image from Microscope...
Done capturing image.
```

Magnification: 20X

Aperture: 0.95

Field Stop: 1.0

Reflector
- ● None
- ○ Analyzer
- ○ DIC RED

Exposure Time: 5ms

Video / Capture
- ● Capture Single Snapshots
- ○ Video Stream

Current Temperature

28.6 C

Temp. Rate (C/min): 10.0

Target Temp. (C): 35.0

Hold Time (sec): 60

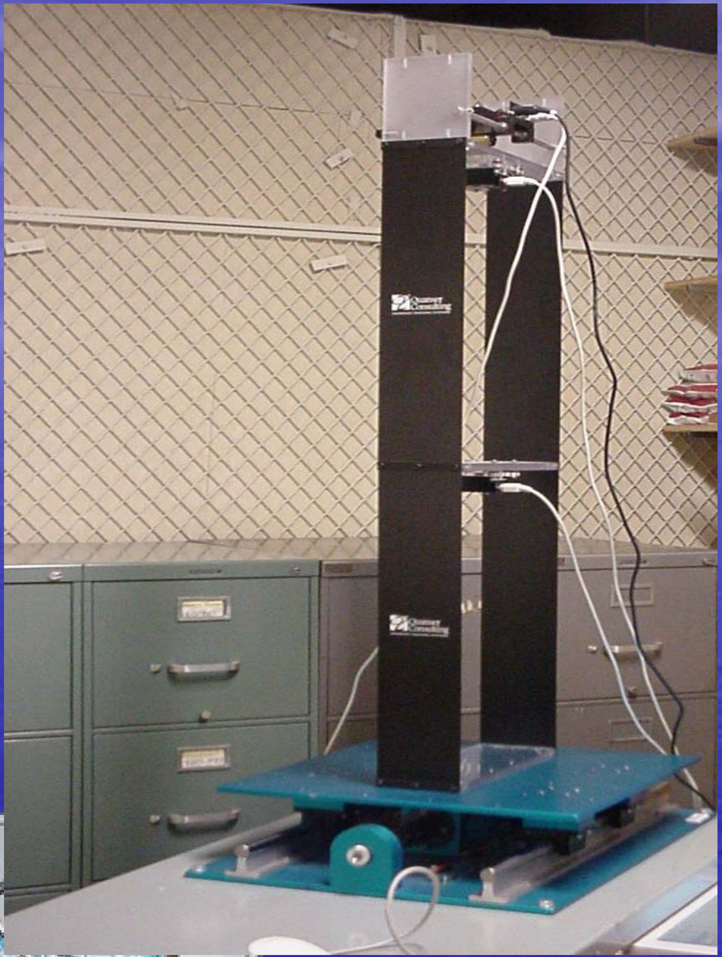
Experiment

mit iCampus

Shake Table

Goal:

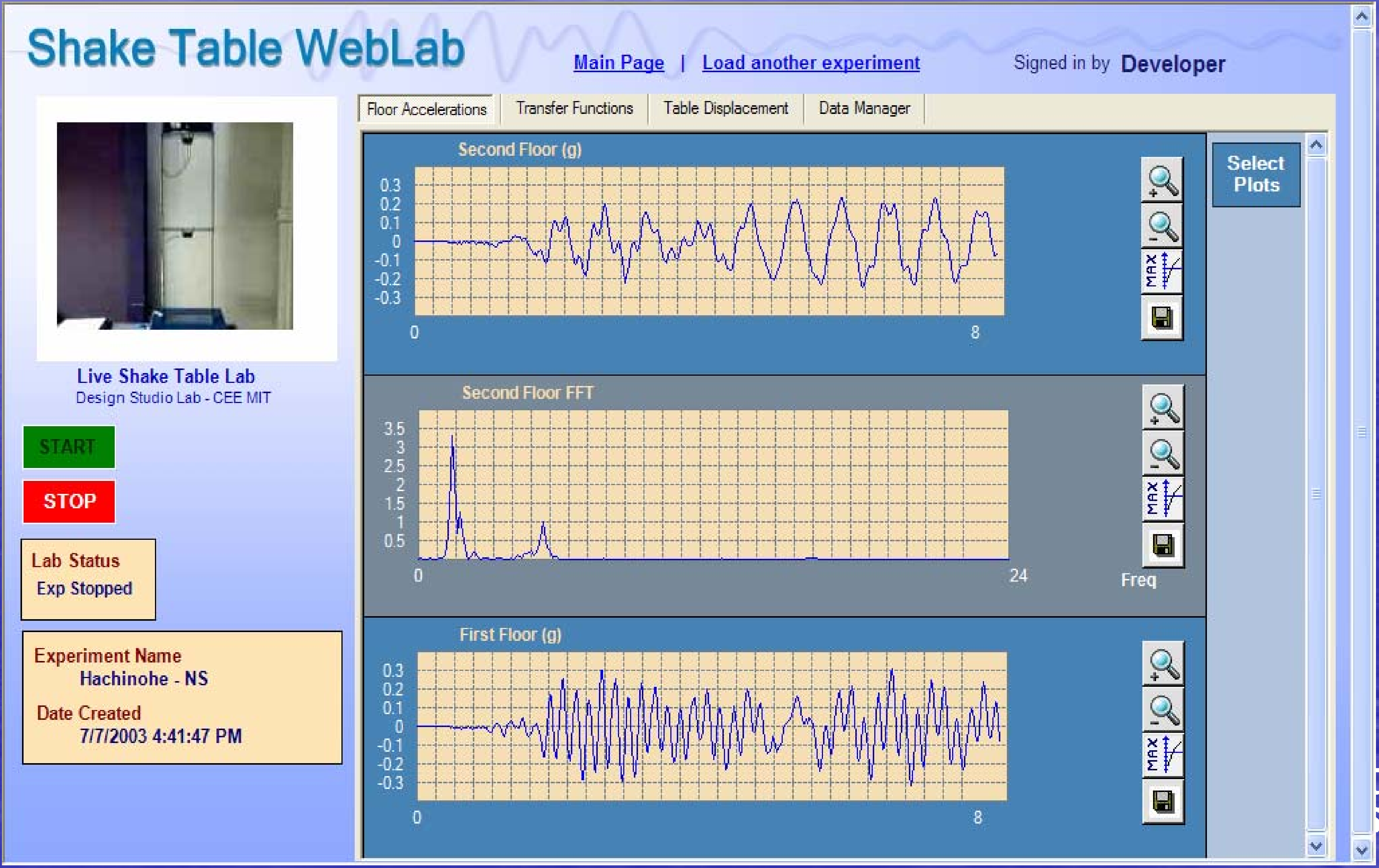
Study behavior of building model structure to ground vibration



Relevance:

Earthquake building engineering

Shake Table GUI



Shake Table WebLab

Signed in by **Developer**

Live Shake Table Lab

Design Studio Lab - CEE MIT

Lab Status

Exp Stopped

Experiment Name

Hachinohe - NS

Date Created

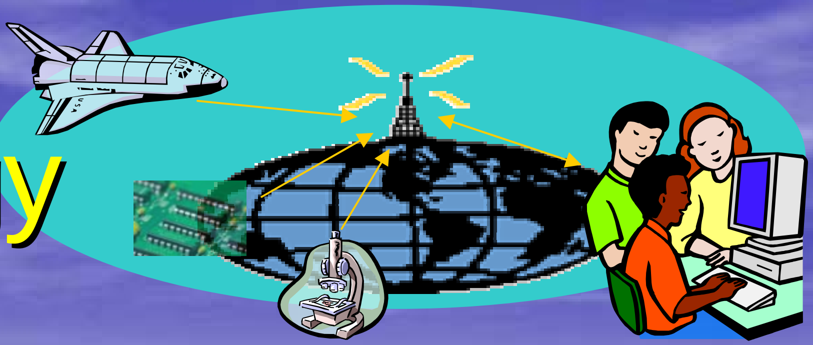
7/7/2003 4:41:47 PM

WebLab: The Student Perspective

- *Students are intrigued and motivated by WebLab*
  - Better student participation and higher scores than regular homework
- *Students dread real laboratories and appreciate WebLab's convenience*
  - Tend to work late at night (unpleasant to be in real laboratory)
  - Simplified interface minimizes frustrations with hardware
  - Can easily work in a "stop-and-go" mode
- *Students have great deal of trouble handling "real-world data"*
  - Can't distinguish good data from bad data
  - Have difficulty manipulating data (graphing, extracting parameters)
  - Have difficulty comparing measured data with theoretical models

mit iCampus

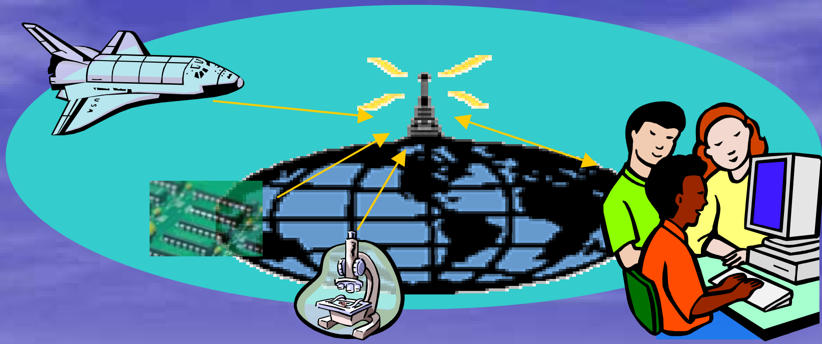
Benefits: Pedagogy



- Create laboratory experiences in subjects that did not have them before.
- Enable laboratory experiments at most opportune moment in curriculum.
- Minimize frustrations with hardware

mit iCampus

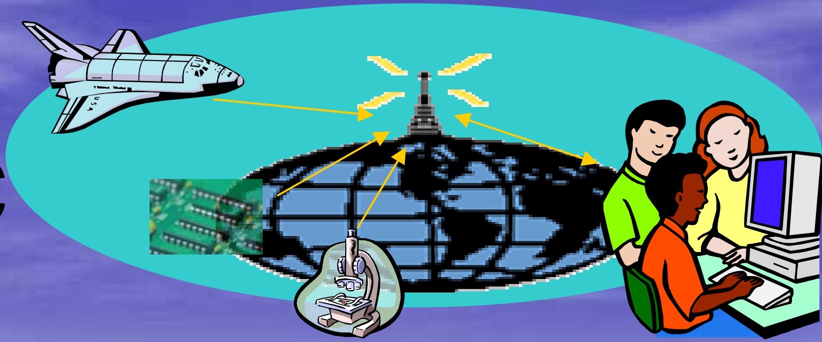
Benefits: Logistical



- Can be located in places inaccessible or hazardous to students
- Allow students to perform experiments in pleasant environments at times of their choice
- Available around the clock
- Accessible from anywhere in the world

mit iCampus

Benefits: Economic



- Share access to expensive equipment
- Costs related to student accessible facilities may be reduced
- WebLabs can be shared

mit iCampus

The iLab Shared Architecture

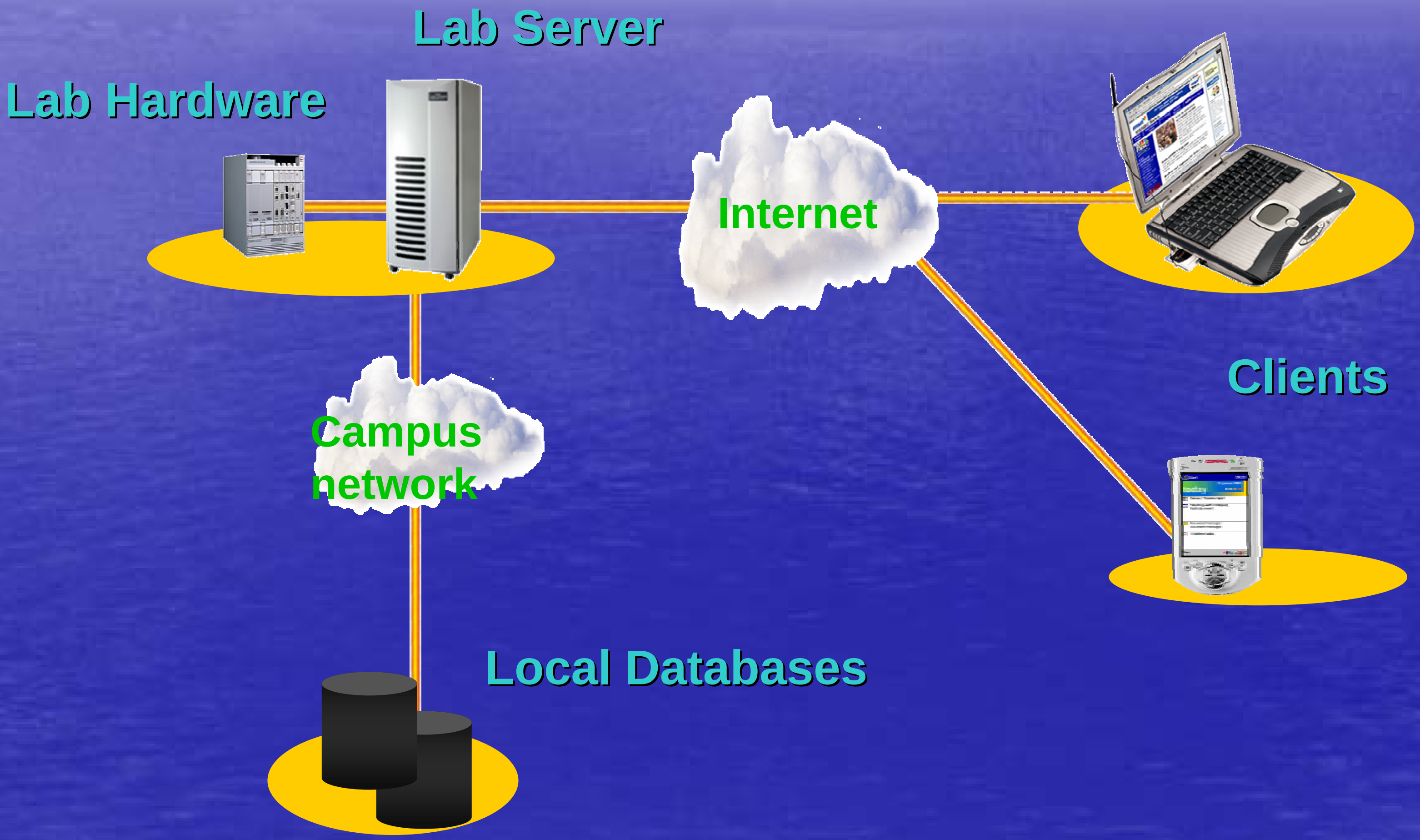
- WebLab Development Obstacles
- iLab Shared Architecture Design Goals
- General Architecture
- Batched Experiment
- Interactive Experiment
- Current Status
- Public Service Broker
- Access to Code

WebLab Development Obstacles

- First Generation WebLab Architecture
- Scalability
- What a Lab Provider Does Not Want To Do

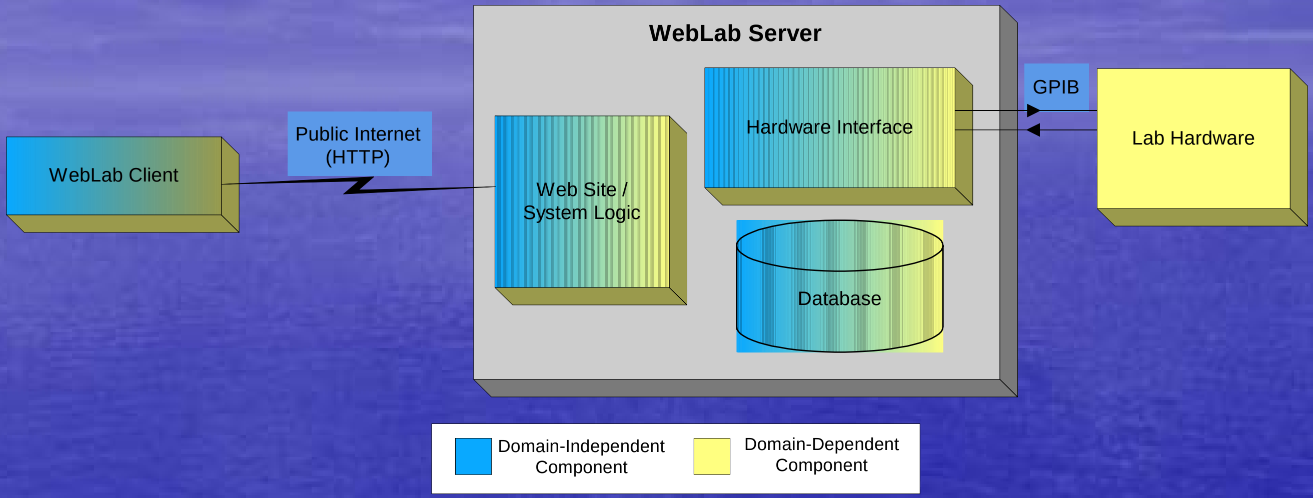
mit iCampus

Typical WebLab Architecture



First-generation WebLabs

(i.e. MicroElectronics WebLab up to v. 5)



- Domain specialist responsible for 100% of development:
  - long time to deployment
- Components are interwoven:
  - complex to debug
  - limited scalability (in terms of features and capacity)
  - components cannot be reused
- Lab manager responsible for entire operation
  - Lab management
  - User registration and data maintenance

WebLab: Scalability

- Each user must be registered on the Lab Server
- Lab manager responsible for User Management and experiment storage policies
- Data storage requirements escalate as the number of users grow.
- Database management
- Server capacity

What a Lab Provider Does Not Want To Do

- Register 100’s of student accounts for other people’s students.
- Store experiment results for students from other institutions and decide when they can be deleted or how to archive them.
- Decide who can view whose experiment results, especially when it involves setting policy for another university’s courses.

mit iCampus

iLab Shared Architecture Design Goals

- iLabs Design Strategy
- iLab Design Goals
- iLab Architecture Boundaries

iLabs Design Strategy

Separate responsibilities of the lab provider from those of the teaching faculty

- The lab provider designs and makes the laboratory experiment available online in as effective a presentation as possible
- The teaching faculty register their own students, manage their accounts and experiment data storage, and set course policy (e.g. can students collaborate)

mit iCampus

iLab Design Goals

- Encourage researchers and universities to share their labs online
- Free lab owner/operator from administration (i.e. authentication, authorization, storage of results, archiving of data, etc.) of users from other universities
- Provide single sign on to labs hosted at multiple universities
- Allow universities with diverse network infrastructures to interoperate and share resources

mit iCampus

iLab Architecture Boundries

- Our architecture does not deal with specific hardware and software interfaces to lab equipment
- Our architecture is intended to be compatible and complementary with commercial software such as National Instrument’s LabVIEW and analysis packages like Matlab

General Architecture

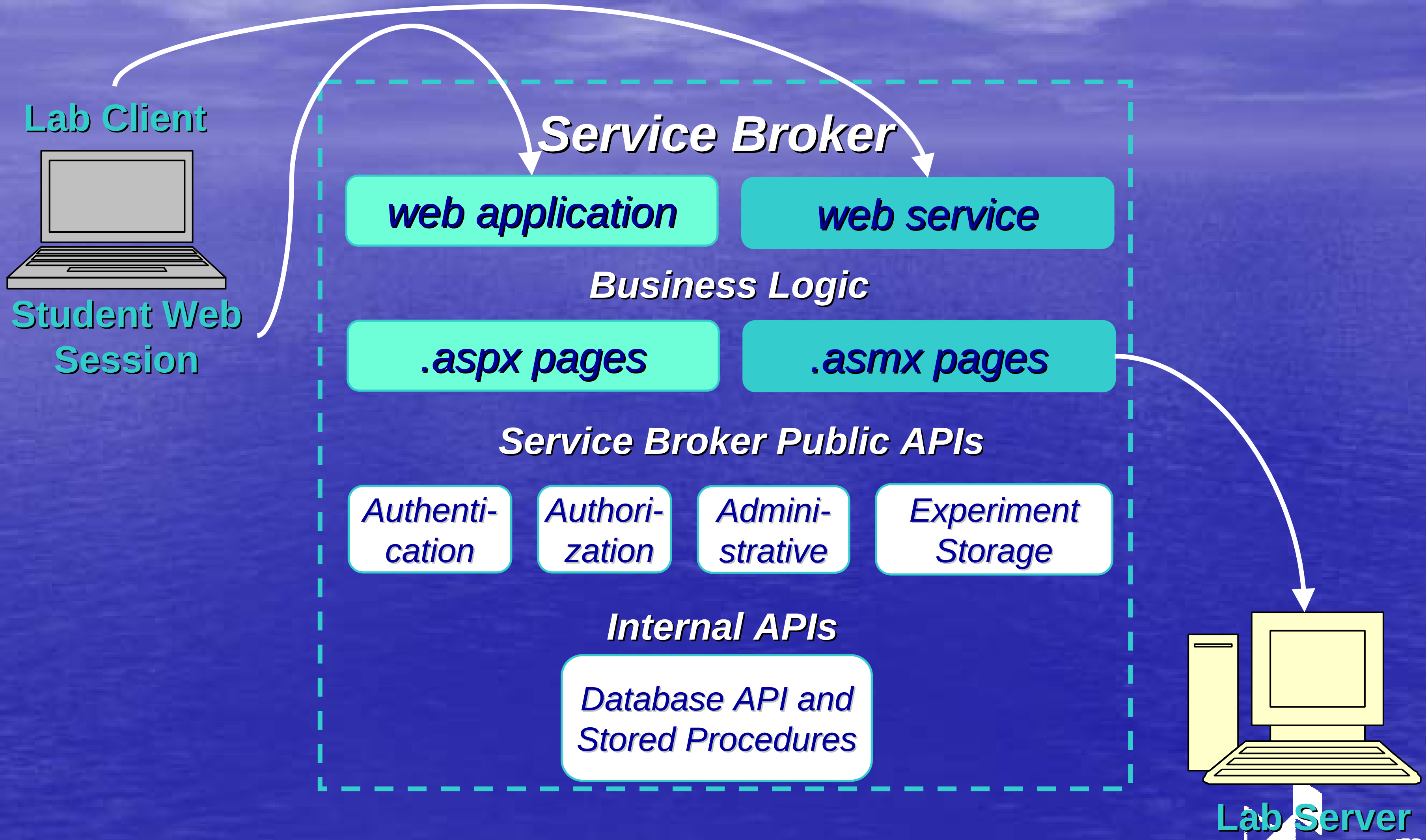
- Access to all services is controlled through the Service Broker.
- Shared code base to provide generic functionality required by most Labs
- Consists of a collection of generic services.
- Builds on top of the current generation of Web Services.

iLab Generic Services

- User authentication (and registration)
- User authorization and credential (group) management
- Experiment specification and result storage
- Lab access scheduling

mit iCampus

iLab Service Broker



Lab Client

Student Web Session

Service Broker

web application

web service

Business Logic

.aspx pages

.asmx pages

Service Broker Public APIs

Authentication

Authorization

Administrative

Experiment Storage

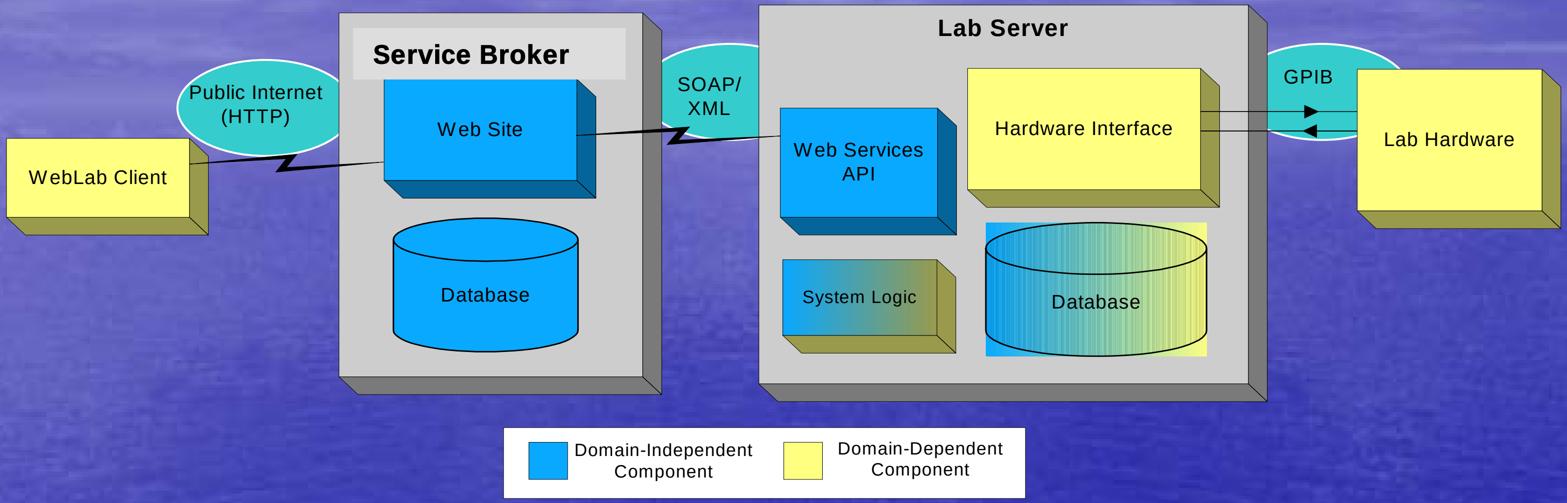
Internal APIs

Database API and Stored Procedures

Lab Server

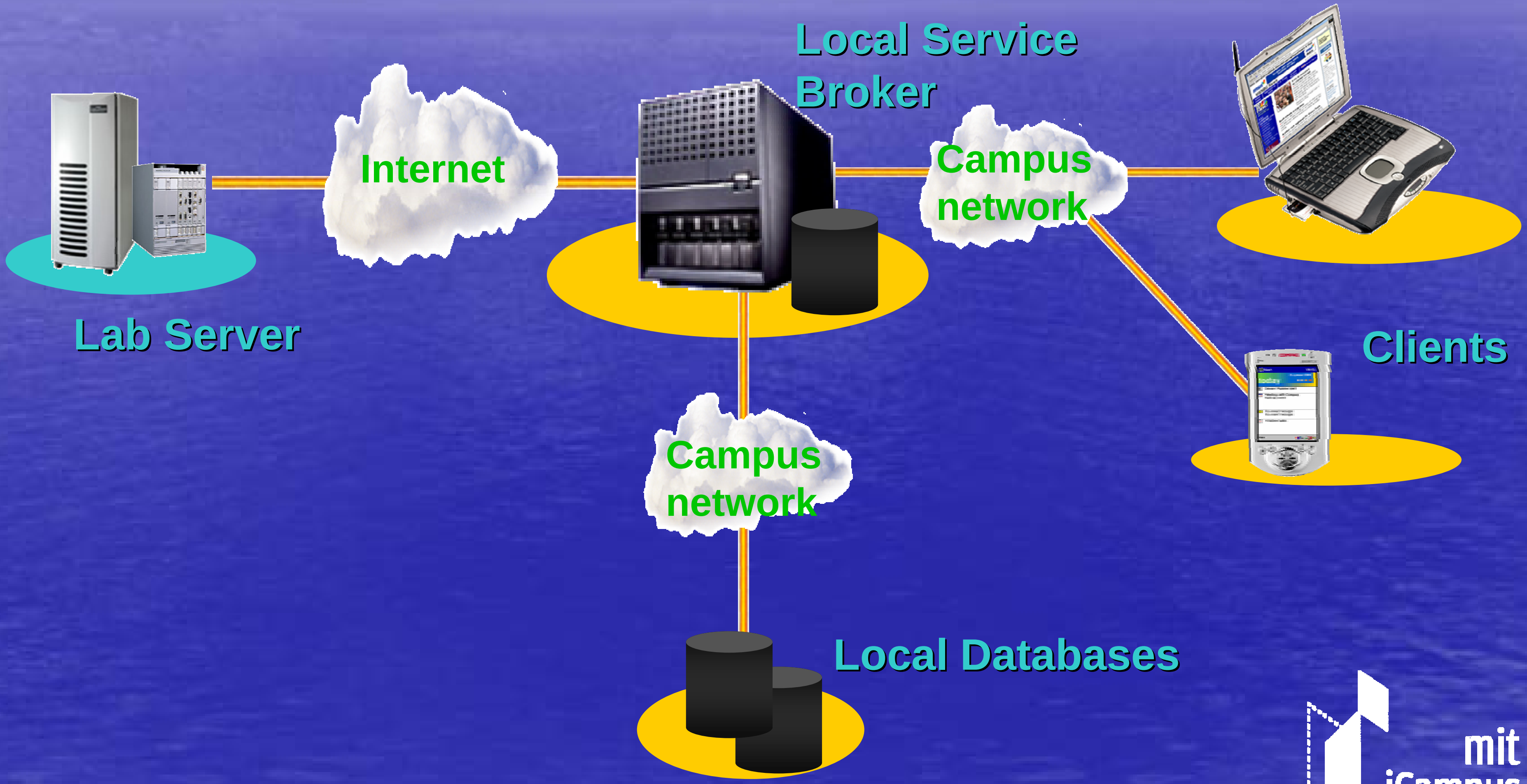
mit
iCampus

iLab Shared Architecture



- iLab components are abstracted from one another
  - modular, reusable
- Domain specialist leads lab development, but:
  - responsible for *domain-dependent* components
  - uses Shared Architecture for generic components
  - Relegates user management to Service Broker
- Enhanced scalability

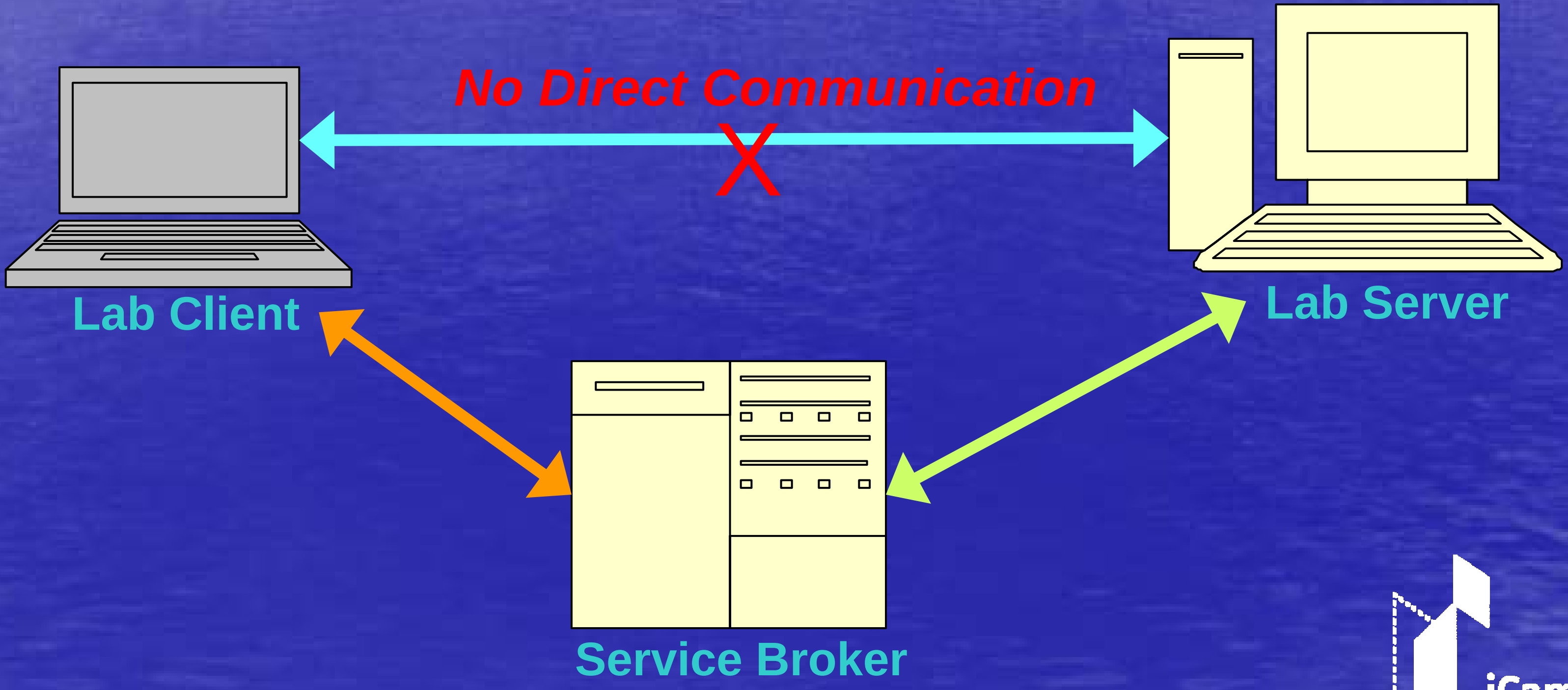
iLab Shared Architecture



mit iCampus

Batched Experiment Network Topology

In the batched experiment architecture, the client and the lab server communicate only through the Service Broker:

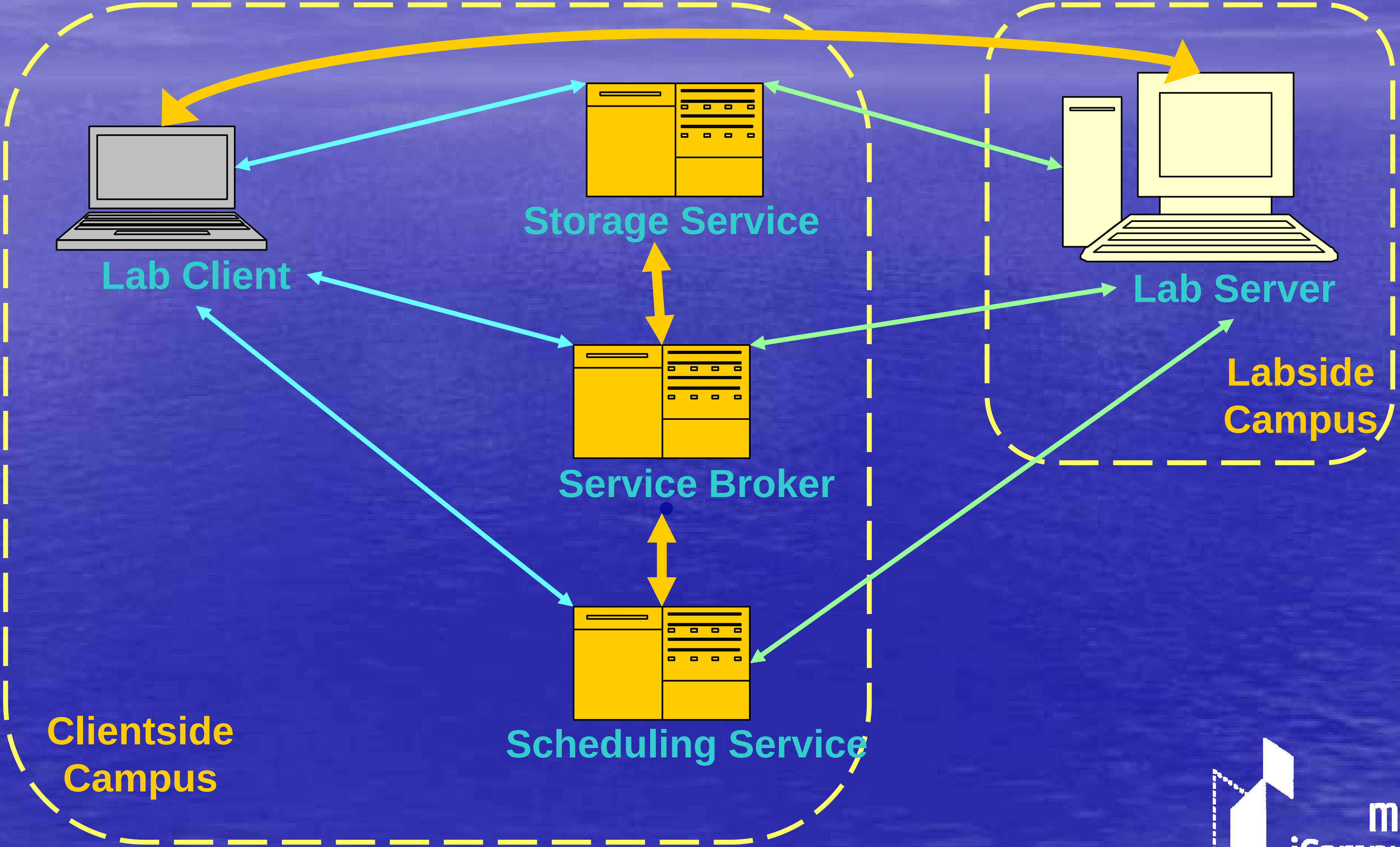


Interactive Experiment Network Topology

In the interactive architecture, the client and lab server will be able to communicate directly:

- To reduce network latency;
- To permit the use of virtual instrumentation toolkits like LabView and other development packages;
- To permit streaming data from lab server to client.

Preliminary Interactive Topology



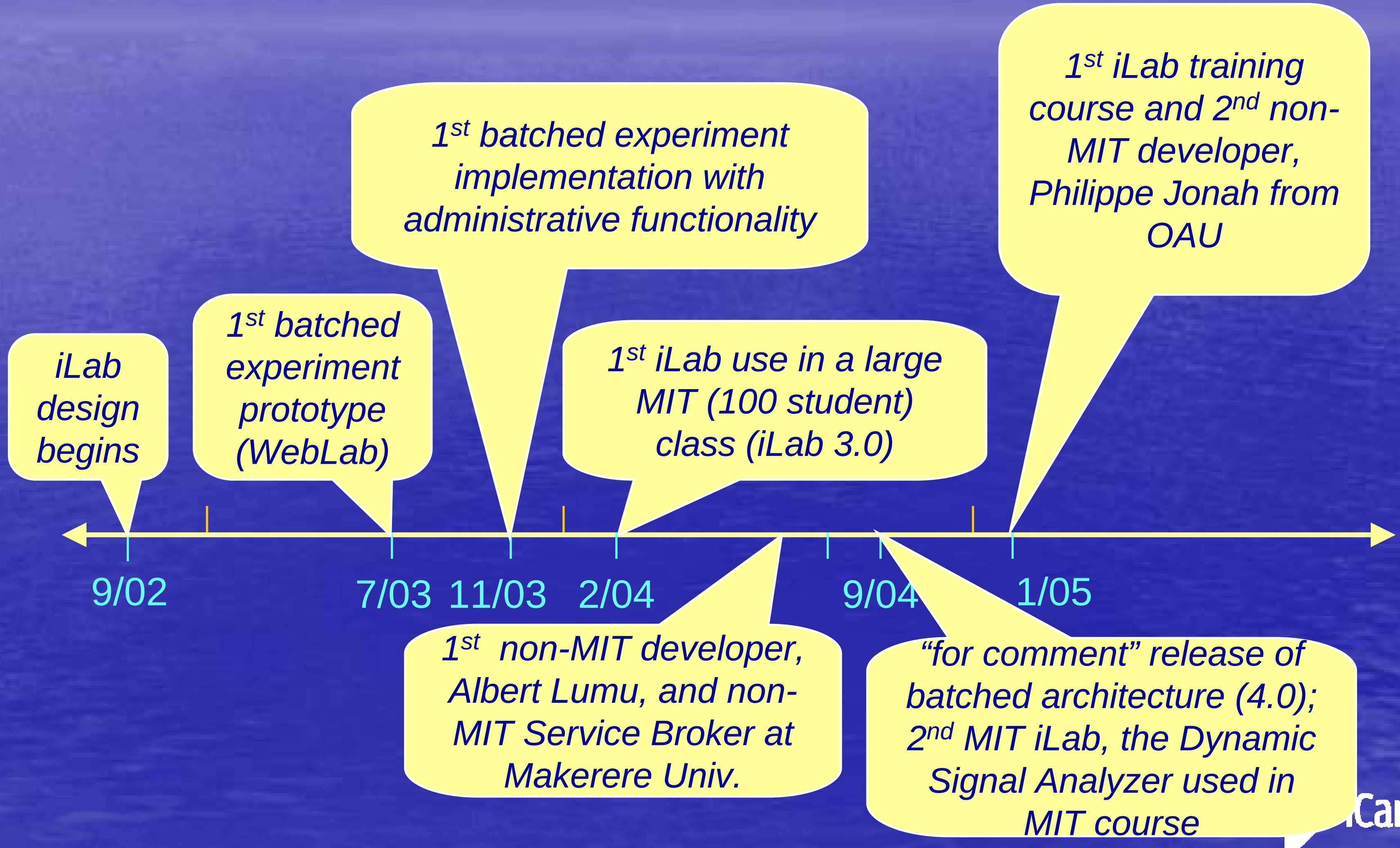
mit iCampus

Current Status

- Timeline
- Collaborations
- openiLabs – a public Service Broker
- Batched Experiment Software Release
- Interactive Architecture

iLab Shared Architecture:

Project Timeline, 1



iLab design begins

1st batched experiment prototype (WebLab)

1st batched experiment implementation with administrative functionality

1st iLab use in a large MIT (100 student) class (iLab 3.0)

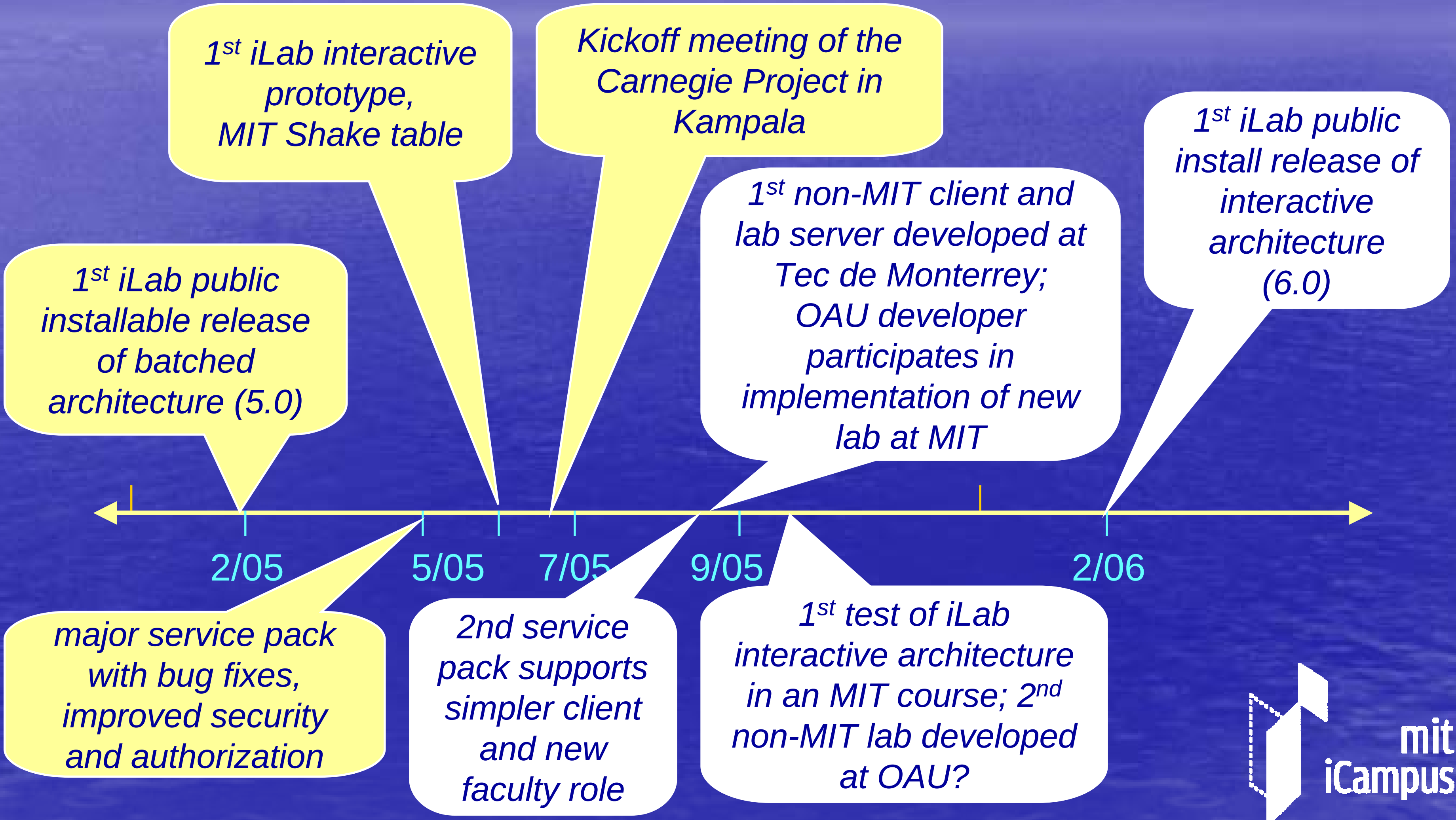
1st non-MIT developer, Albert Lumu, and non-MIT Service Broker at Makerere Univ.

"for comment" release of batched architecture (4.0); 2nd MIT iLab, the Dynamic Signal Analyzer used in MIT course

1st iLab training course and 2nd non-MIT developer, Philippe Jonah from OAU

iLab Shared Architecture:

Project Timeline, 2



1st iLab public installable release of batched architecture (5.0)

1st iLab interactive prototype, MIT Shake table

Kickoff meeting of the Carnegie Project in Kampala

1st non-MIT client and lab server developed at Tec de Monterrey; OAU developer participates in implementation of new lab at MIT

1st iLab public install release of interactive architecture (6.0)

major service pack with bug fixes, improved security and authorization

2nd service pack supports simpler client and new faculty role

1st test of iLab interactive architecture in an MIT course; 2nd non-MIT lab developed at OAU?

mit iCampus

Collaborations

- We are promoting iLabs to create a larger community of WebLab developers and users.
- We are actively working with universities in Singapore, Sweden, China, Mexico, Taiwan, Lebanon, Uganda, Nigeria, Tanzania, Italy, Colombia, and Greece on sharing iLab experiments.
- Regional Service Brokers have been established in Nigeria, Uganda, Mexico and China.
- The MicroElectronics WebLab had over 1,000 users during the Fall 2004 semester.
- MIT iCampus Outreach has established an affiliated institution program.

openiLabs – a public Service Broker

- In October 2004 established a free, public Service Broker – http://openilabs.mit.edu
- Goal is to allow faculty to experiment with use of online labs without committing to implementing them on their campus
- People can automatically create accounts from outside MIT on this Service Broker for themselves and their students

mit iCampus

Batched Experiment Software Release

- In February, 2005, we released the first version of the Batched Experiment Service Broker SDK.
- A bug reporting, tracking and fixing process is in place.
- Get the latest version: http://icampus.mit.edu/ilabs/architecture

mit
iCampus

Service Broker SDK includes:

- White papers describing overall architecture and design philosophy
- Specifications of web service APIs
- Full code and database scripts for building the Service Broker
- Detailed “read me” file describing build procedures
- Simple example of lab client and lab server (equivalent of “Hello world” program)
- Full code for Microelectronics Weblab Version 6 (lab client and server)

iLab Intellectual Property Policy

- All MIT developed iLab software has been and will continue to be made available for free under an open source license.
- We encourage but do not require our academic partners to follow the same policy. The decision to share their code and under what terms is their to determine.
- We allow industrial partners to develop commercial “shrink-wrapped” (supported) versions of the iLab components.

mit iCampus

Interactive Architecture Timeline

- June 2004: Polymer Crystallography prototype
- June 2005: alpha release
- September 2005: trial lab in course at MIT
- January 2006: First full release

mit
iCampus

The iLab Vision for the Future

- Projected iLab Milestones
- Interoperability with Commercial Lab Software
- Future Goals for Web-based Labs

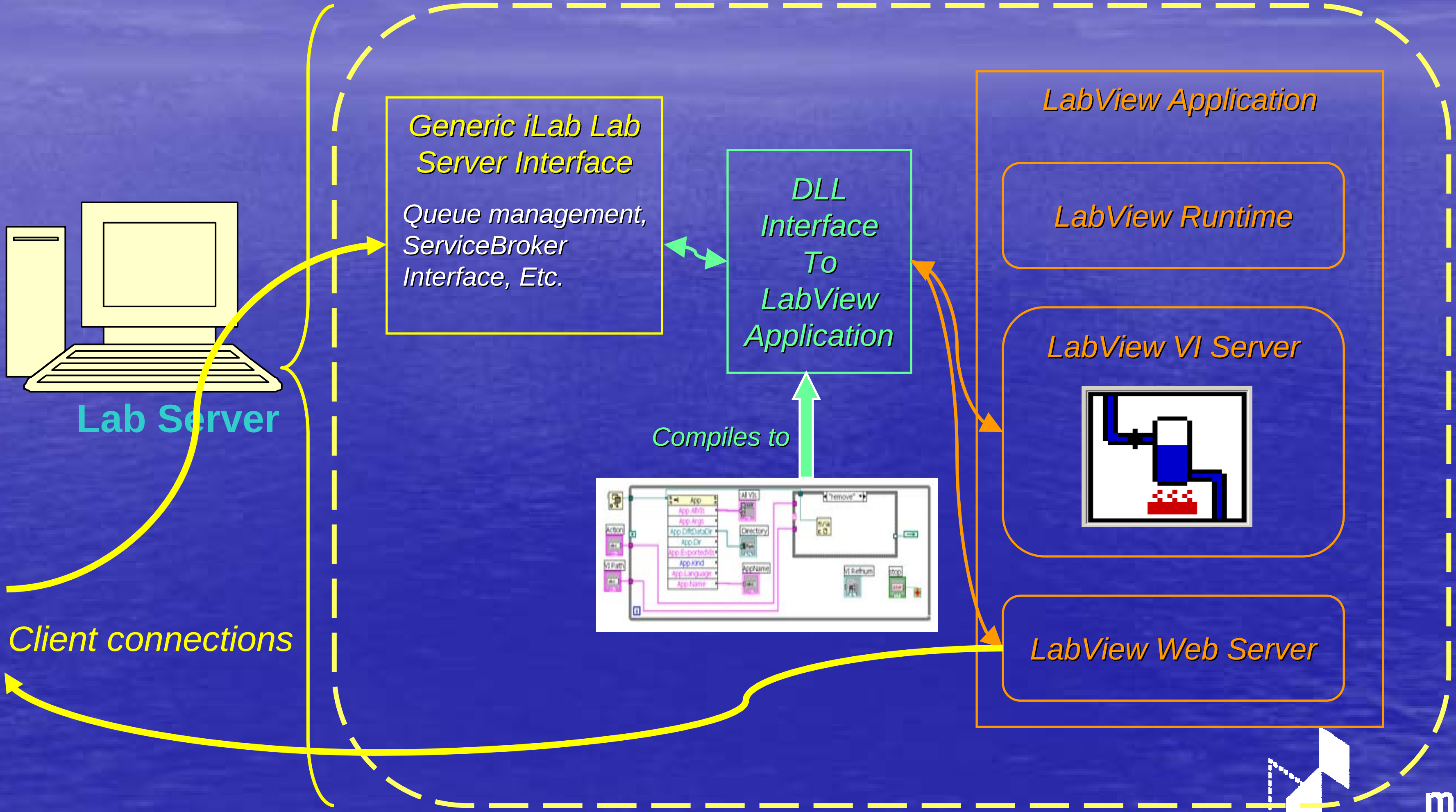
Projected Project Milestones

- Spring/Summer 2005 – implementation and testing of APIs for interactive experiments
- Early Fall 2005 – release of interactive experiment specifications and code for comment
- Early Fall 2005 – release of prototype LabVIEW enabled Lab Server, with guidelines for converting existing applications.
- Jan-Feb 2006 – full release of interactive experiment code, documentation, "how to" manual and other materials
- 2006-2007 – repeat cycle for sensor-based experiments

Interoperability with Commercial Lab Software

- Many universities already use commercial lab software, notably LabVIEW
- We have ongoing discussions with LabVIEW developers and management
- Goal is interoperability, not competition
- iLab can potentially add important functionality to commercial products
- Prototype development is in process

Preliminary iLab-LabVIEW Architecture



Lab Server

Generic iLab Lab Server Interface

Queue management, ServiceBroker Interface, Etc.

DLL Interface To LabView Application

Compiles to

LabView Application

LabView Runtime

LabView VI Server

LabView Web Server

Client connections

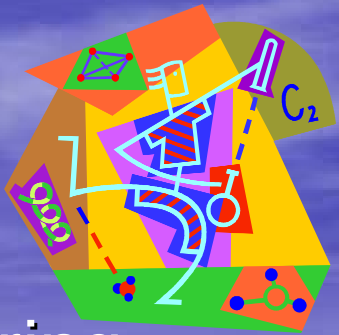
mit iCampus

Future Goals for Web-based Labs

- Improving education through expansion of lab-based learning opportunities around the world
- Creating a movement within higher education (and potentially other levels) leading to global sharing of laboratory experiments over the net
- Sharing beyond access to lab equipment to include pedagogical materials and teaching experiences
- Creating an informal “barter economy” to facilitate sharing of lab equipment
- “iLab-ready” experimental equipment and software
- Sharing of time on national and international experimental equipment such as space-based experiments

mit
iCampus

Conclusions



- iLabs will enhance science and engineering education
- iLabs and their educational content will be broadly shared around the world
- iLabs provide a path for the developed world to support the educational objectives of the developing world
- iLabs Shared Architecture: a scalable framework for WebLab development and management

Contact Information

- openilabs.mit.edu – The public Service Broker
- icampus.mit.edu/iLabs – iCampus Outreach site for iLabs
- icampus.mit.edu/iLabs/architecture – Download project code, presentations and documentation. Access to on-line iLab forum.

- Philip H. Bailey – pbailey@mit.edu