

# Developing Interactive iLabs

1<sup>st</sup> iLab Europe Workshop  
November 16 - 18, 2009



# Developing Interactive Labs for the ISA

---

By design, few technical constraints are placed on lab development

- ▶ Labs can have drastically different requirements
  - ▶ Communication protocols
  - ▶ Could be based on proprietary technologies (between client and lab server)
- ▶ One size does not fit all
- ▶ Regardless, a few rules apply to building Interactive iLabs



# Supporting ISA Services

As with Batched iLabs, Interactive iLabs must implement iLab control API's

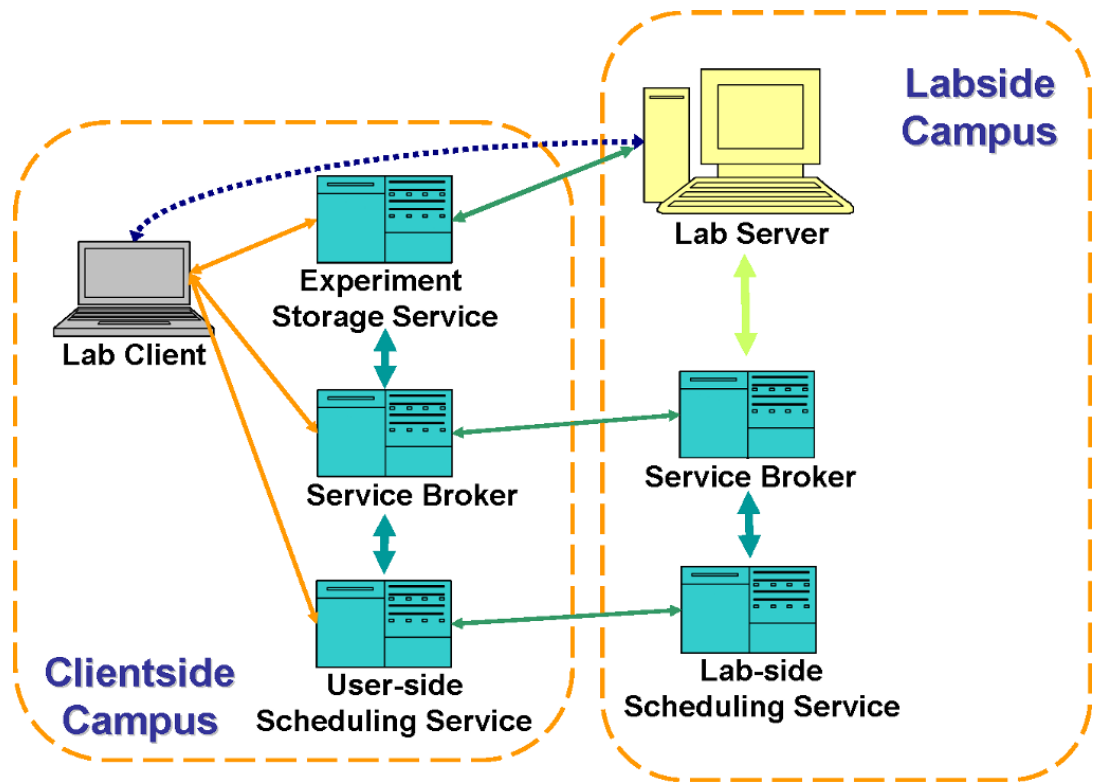
- ▶ Control API's based on Web Services

- ▶ Lab Server:

  - ▶ Service Broker, ESS, Scheduling, Ticketing

- ▶ Lab Clients

  - ▶ Service Broker, ESS, Ticketing



# Direct Connections to iLab Services

---

Some services are required, others are optional

- ▶ Core Service Broker and Ticketing interfaces – Required
  - ▶ Support for Ticketing necessary for ISA authentication
  - ▶ Support for Service Broker needed for ticket redemption
- ▶ Support for Storage, Scheduling Services optional, can be done in different ways
  - ▶ Must only be implemented if required by lab.
  - ▶ For Storage, responsible party depends on specific lab
    - ▶ Lab Server for high-bandwidth or large file storage
    - ▶ Client for situations where user analysis is part of lab record
    - ▶ Hybrid models



# Technology Decisions

---

Lab Server, Client platforms driven primarily by instrumentation, pedagogical requirements

- ▶ What are the requirements of our lab hardware?
  - ▶ Driver availability
  - ▶ Supported platforms
- ▶ How do we want the lab to be used?
  - ▶ Instrument control
  - ▶ Data formats, modes of analysis,



# Supported Lab Client types

---

Wide variety of client interfaces/technologies available, but only a limited set supported currently

- ▶ Java Applet/Web Start programs
  - ▶ Similar to MIT batched iLab clients
  - ▶ Run as distinct processes on user's computer, must support web services for ticketing, experiment storage
  - ▶ Can use any protocol to communicate with Lab Server
- ▶ HTTP Redirects
  - ▶ Supports wide range of clients hosted as web-based UI's backed by software on Lab Server
  - ▶ Technically, Client and Lab Server are connected
  - ▶ HTML Forms, Flash-based clients, LabVIEW VI's
- ▶ Other mechanisms possible, support depends on existing use cases



# Supporting LabVIEW-based iLabs

---

Support for LabVIEW-based iLabs has become a high priority for the MIT iLab Project

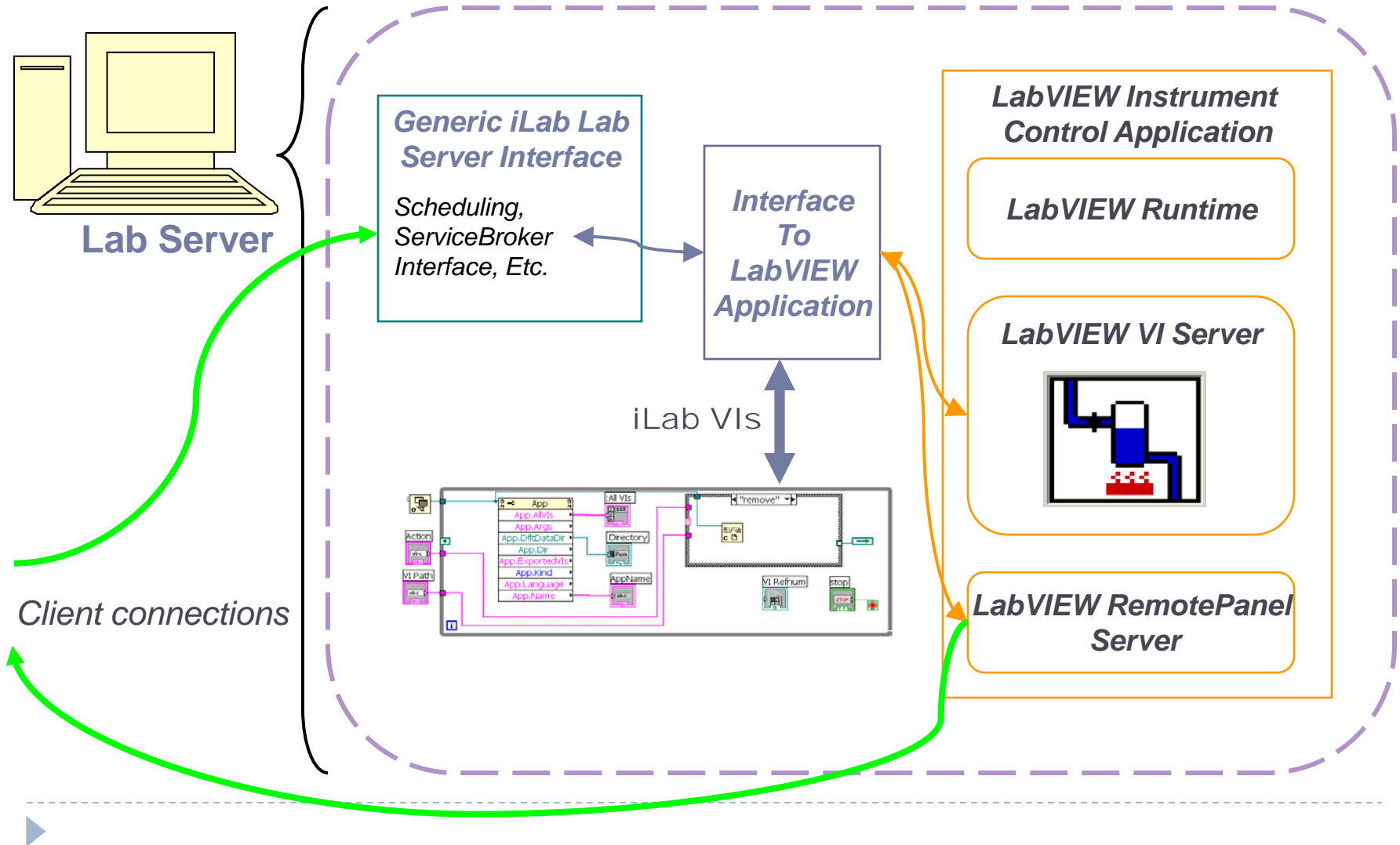
- ▶ Many adopters already use LabVIEW for instrument control
- ▶ Can easily control instrumentation, develop intuitive user interfaces
- ▶ Low barrier to entry for non-software engineers

In response to this, we have developed a toolkit for streamlining conversion of a working VI to an Interactive iLab

- ▶ LabVIEW Integrated Interactive Lab Server (LVIILS)



# LabVIEW Integrated ILS Architecture





# Using the LVIILS

---

The LabVIEW Integrated ILS is packaged with the current Service Broker software, enables:

- ▶ Easy installation of general purpose Lab Server code, integration with ISA Services
- ▶ Existing LabVIEW VIs can take full advantage of ISA Services
  - ▶ VI's must conform to a few requirements
- ▶ A working instrument control VI can be converted to a complete Interactive iLab in a short amount of time
  - ▶ Easier, quicker deployments of fully-featured iLabs



# Support for other Interactive Labs

---

- ▶ No other toolkits exist at this time

However...

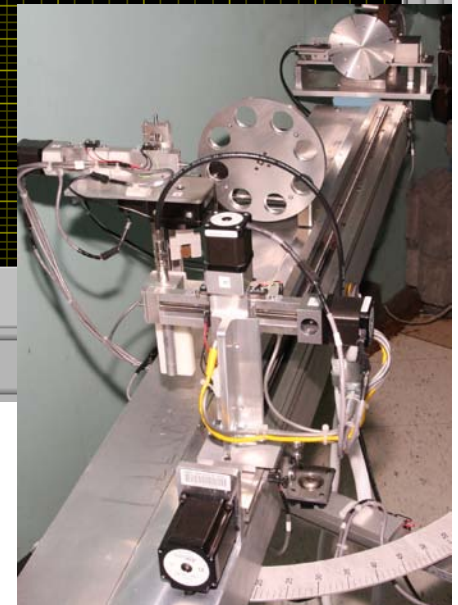
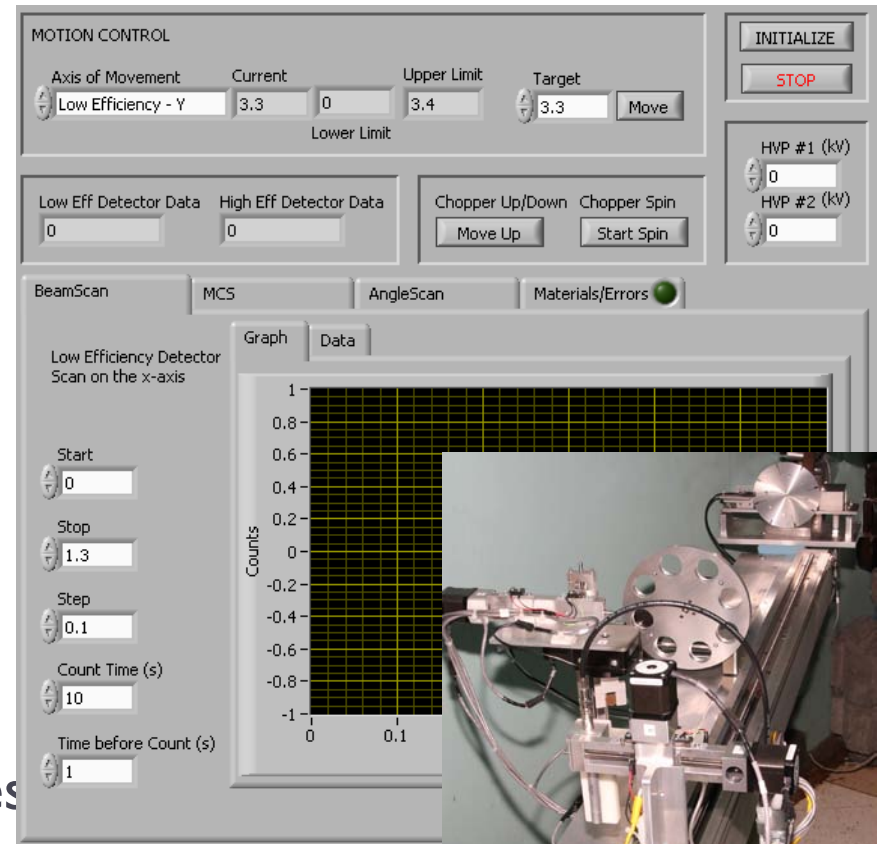
- ▶ Sample Interactive Lab Server packaged with Service Broker software can be used as a starting point for development. Contains:
  - ▶ Experiment launch and monitoring
  - ▶ Experiment termination at end of scheduled run.
  - ▶ Connection management ESS data storage
- ▶ Implementation of LVILS can be used as a development example
- ▶ Support for other platforms driven by user needs



# Deployed Interactive iLabs: Nuclear Reactor

## Exposes some functionality of the MIT Nuclear Research Reactor

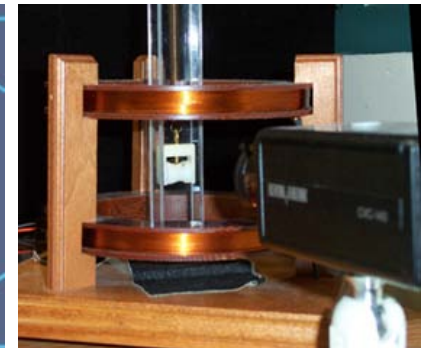
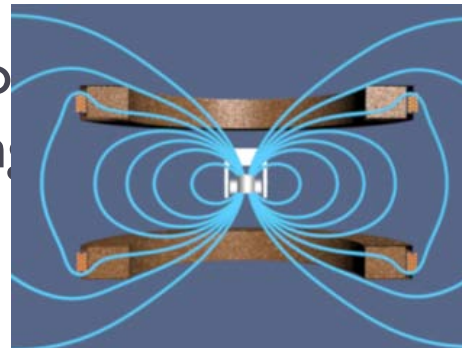
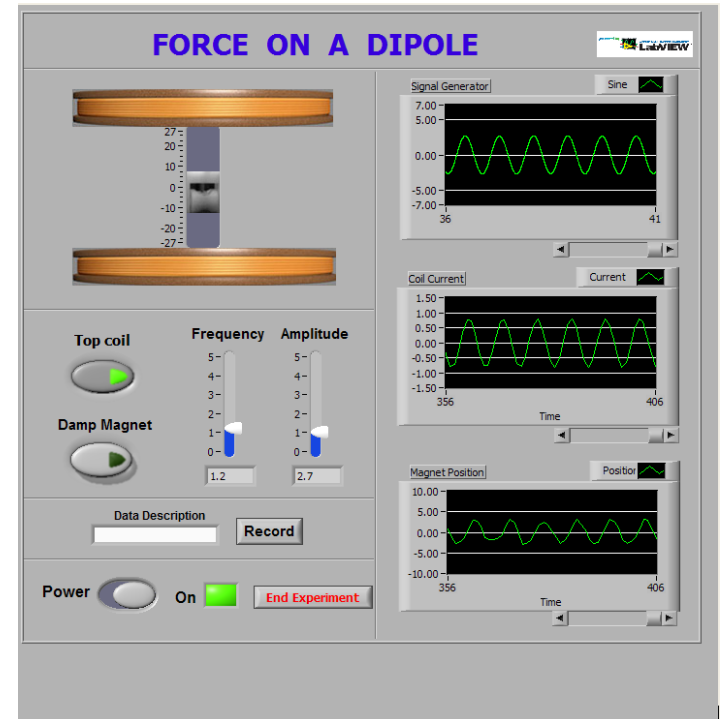
- ▶ A variety of experiments are available in the iLab
- ▶ Also built using LabVIEW and LVIILS
- ▶ To be used in MIT Nuclear Engineering and Physics courses and, eventually, by select secondary schools



# Deployed Interactive iLabs: Force on a Dipole

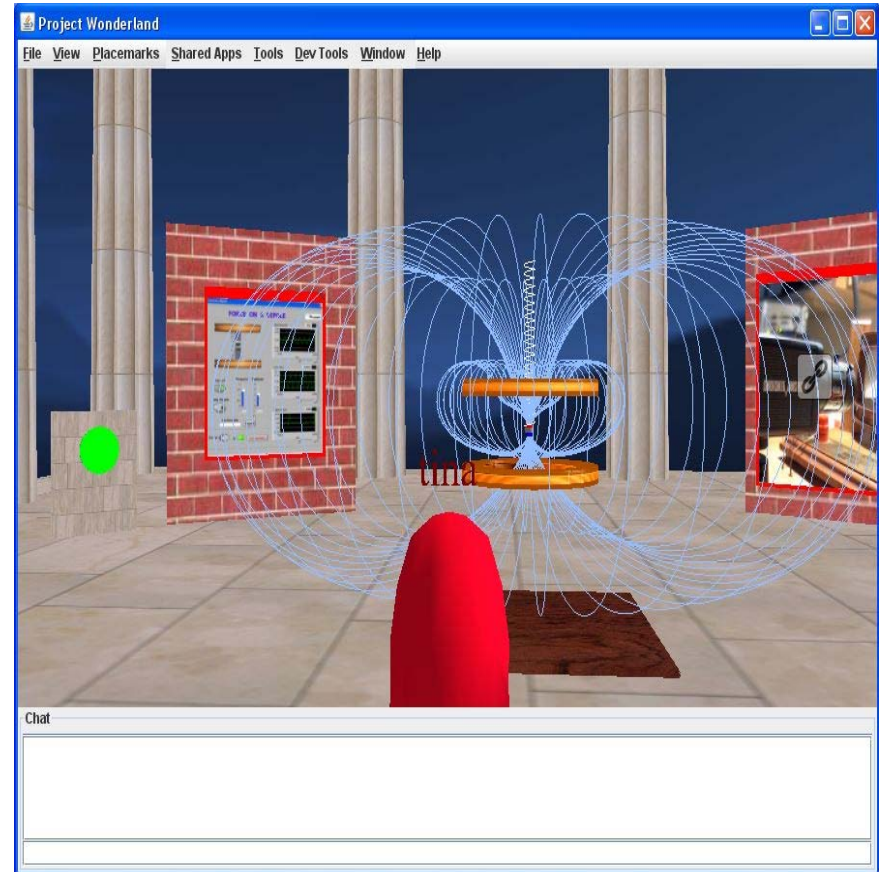
Enables students to observe the behavior of a magnet suspended by a spring between a Helmholtz coil

- ▶ Targeted to freshmen Physics students at MIT
- ▶ Also built using LabVIEW
- ▶ Successful test deployment Spring 2008, full scale deployment (~600 students) this Spring



# iLabs in Virtual Worlds

- ▶ Provides an alternate interface to Force on a Dipole iLab
- ▶ Based on Sun's Wonderland platform (v0.4 for demo)
- ▶ Enables interaction with visualization elements, intuitive collaboration
- ▶ Work continuing
  - ▶ Support for newer version
  - ▶ Integration of other iLabs



**Demo Video**

# Conclusion

---

- ▶ iLab Shared Architecture provides flexible support for wide range of Interactive lab experiences
  - ▶ Varying communication paradigms, technologies, interfaces
  - ▶ Much depends on lab developer requirements
- ▶ Flexibility is double-edged
  - ▶ Good: Wide variety of labs can be supported
  - ▶ Bad: No single clear path for lab development
  - ▶ LVILS is the compromise – and a good start
- ▶ Next: Experiences of other developers

