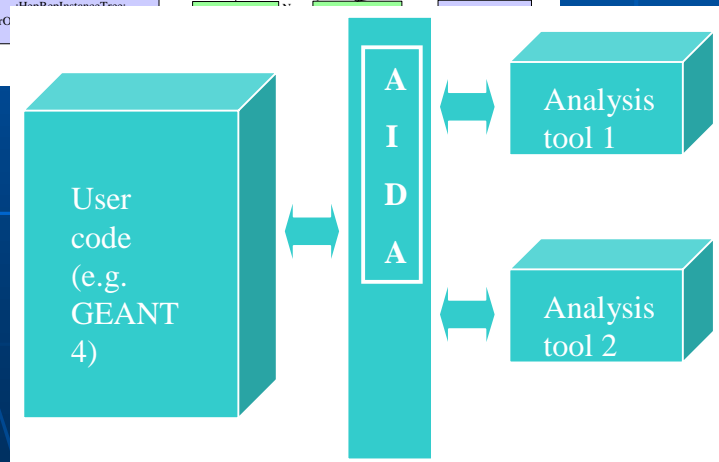
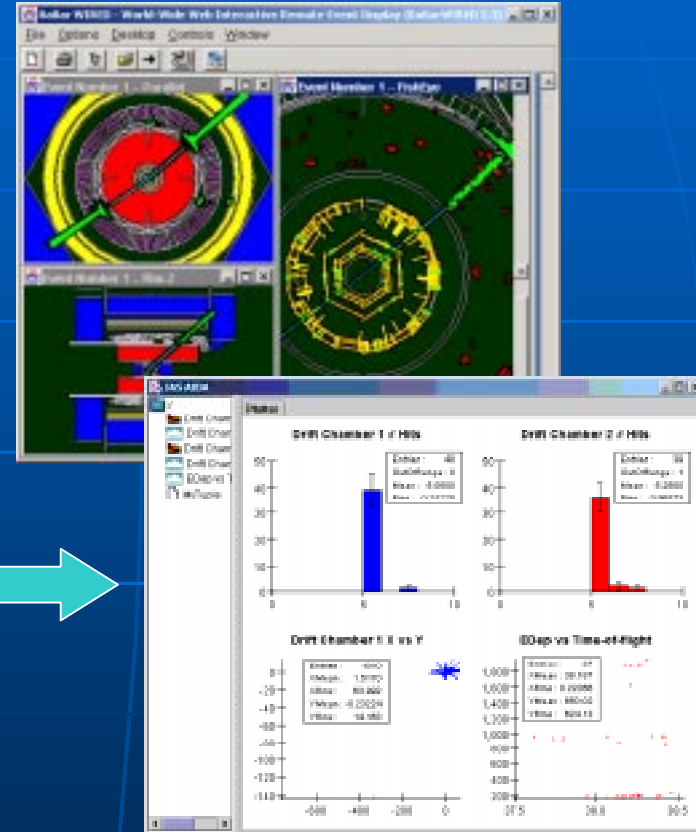
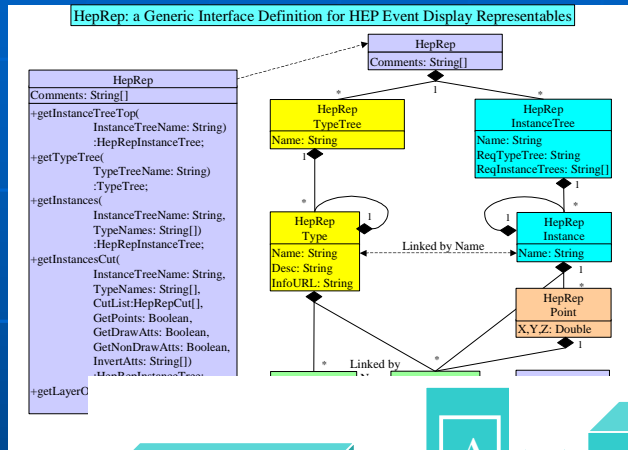


# Visualization of Geant4 Data: Exploiting Component Architecture through AIDA, HepRep, JAS and WIRED



Geant4 Workshop, CERN - 2 October 2002  
Joseph Perl

# Visualization of Geant4 Data: Exploiting Component Architecture through AIDA, HepRep, JAS and WIRED

- Component Architecture/Generic Interfaces
- AIDA for data visualization and analysis
  - Introduction
  - Current use architecture
  - Proposed use architecture
- HepRep for single event visualization
  - Introduction
  - Current use architecture
  - Proposed use architecture
- Demo WIRED reading HepRep files
- Demo JAS reading AIDA files

# Component Architecture / Generic Interfaces

## ■ Component Architecture

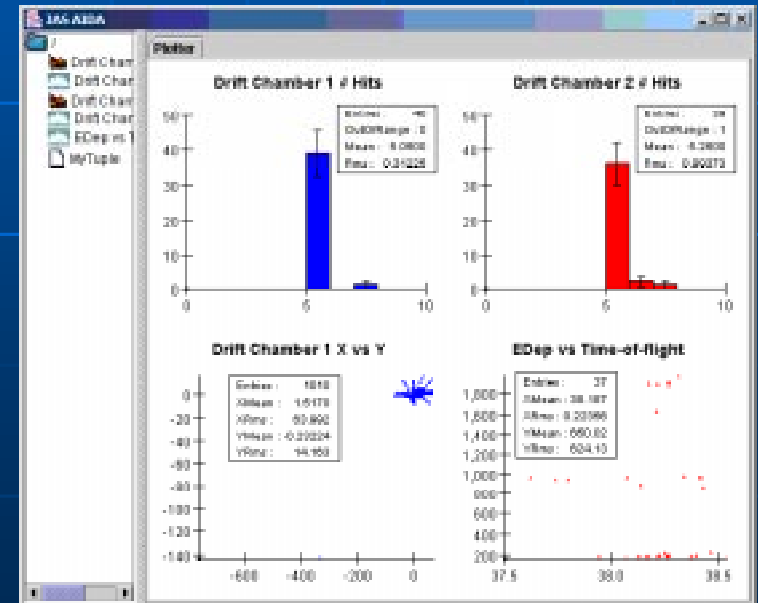
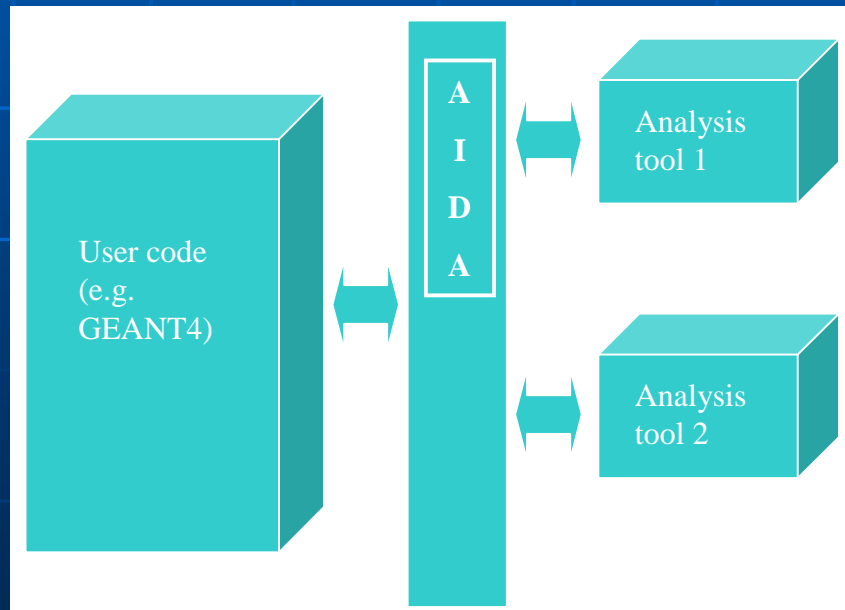
- Use the same data visualization/analysis tool for G4 data as for other experimental data.
- Experiment or collaboration need only agree on the interfaces with which they will fill or send data.
- The end user (physicist, astronomer, doctor) is free to choose whatever desktop tool they prefer.

## ■ Depends on Generic Interfaces

- Well defined (AIDA 3.0, HepRep2)
- Flexible
- Language neutral

# AIDA for Data Visualization and Analysis

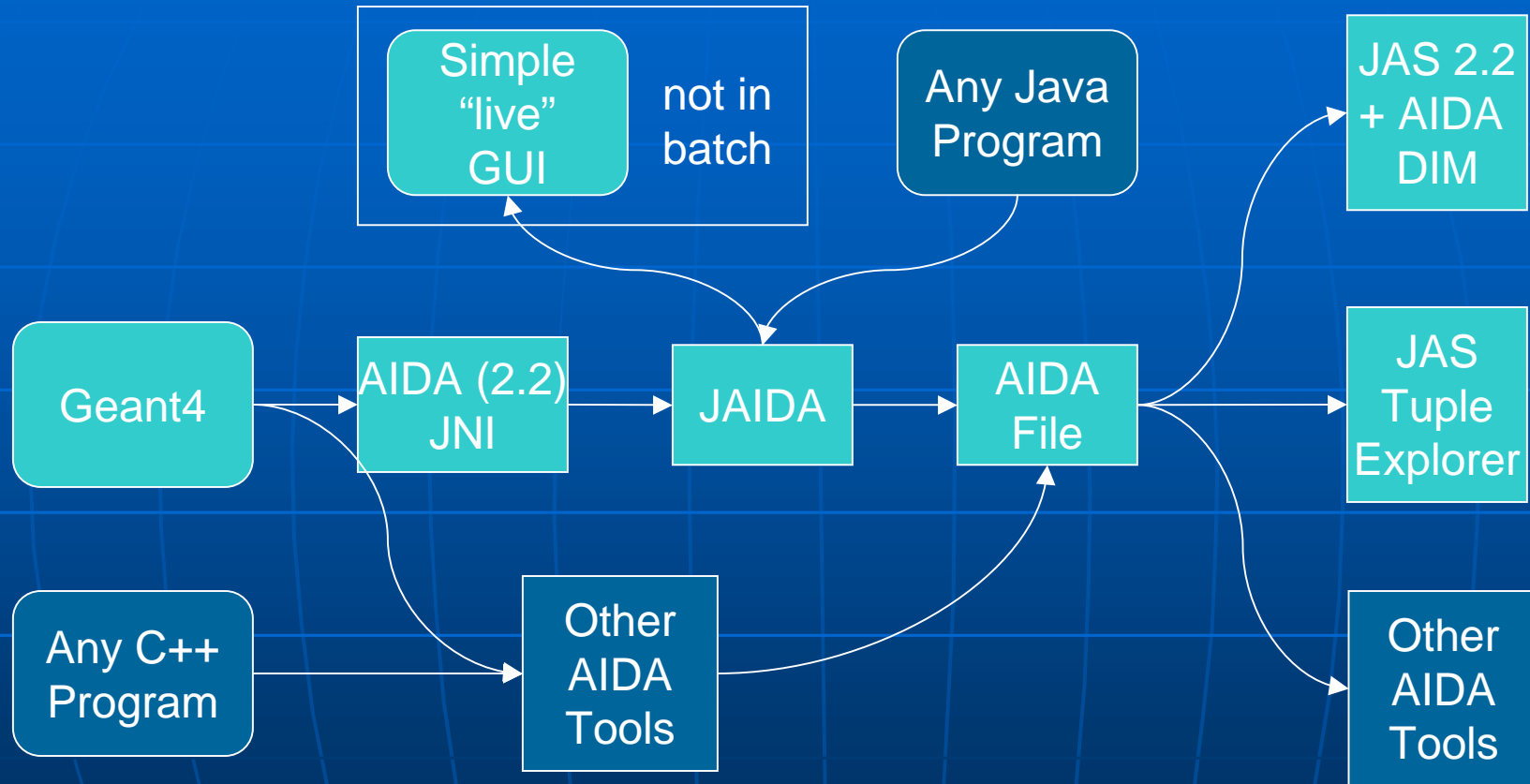
- Abstract Interfaces for Data Analysis (AIDA) is an analysis interface that can be used with several different analysis packages
  - such as
    - JAS
    - Lizard
    - OpenScientist



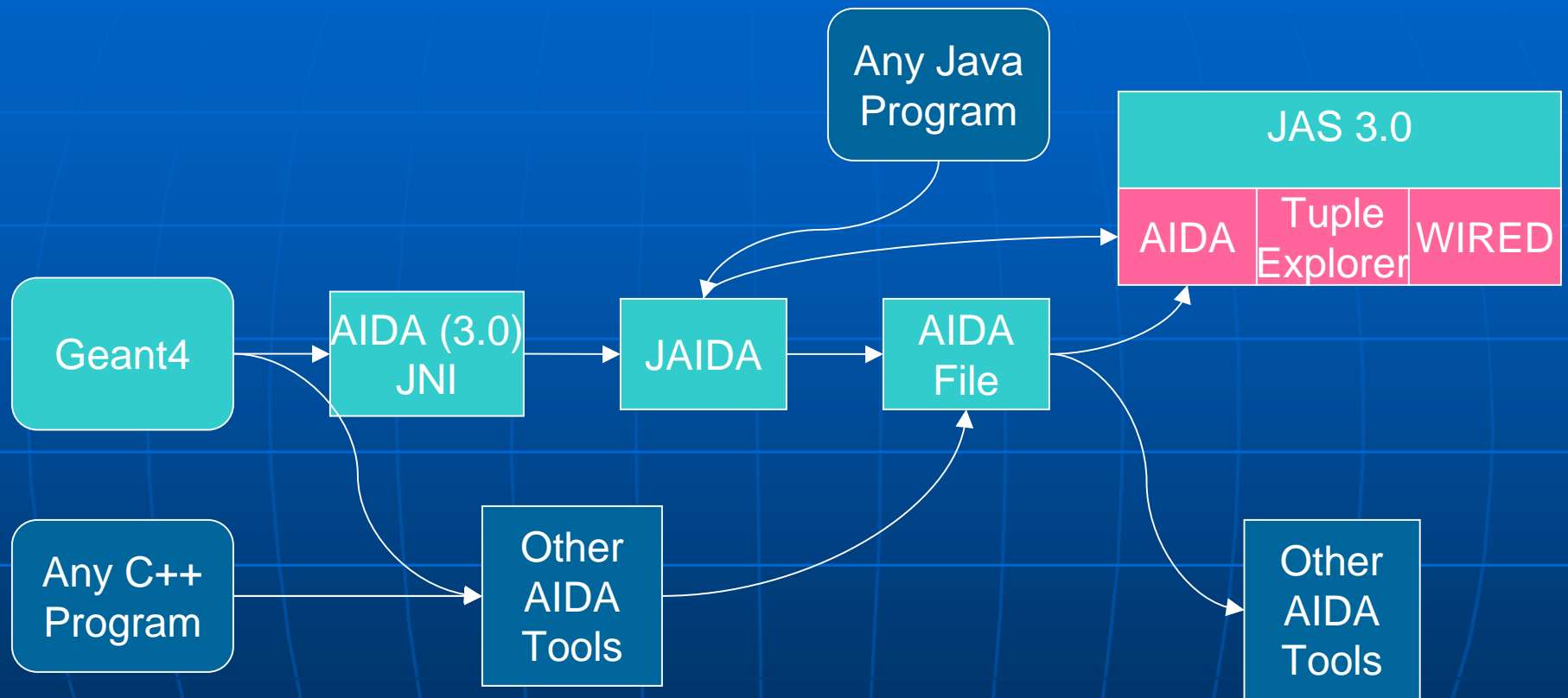
# AIDA Purpose

- *"The goals of the AIDA project are to define abstract interfaces for common physics analysis objects, such as histograms, ntuples, fitters, IO etc. The adoption of these interfaces should make it easier for developers and users to select to use different tools without having to learn new interfaces or change their code. In addition it should be possible to exchange data (objects) between AIDA compliant applications."*

# AIDA Current Use Architecture as used in February Geant4 User Workshop



# AIDA Near-Term Future Architecture



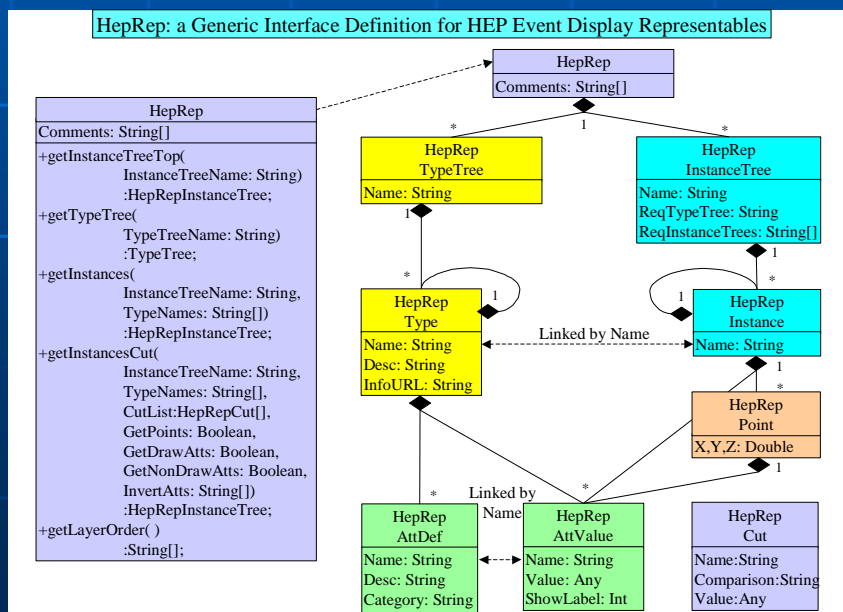
Slightly longer term future:

- GAG style Plug-in for controlling G4
- True interactive analysis (i.e. actually defining histograms on the fly rather than just viewing pre-defined histograms)



# HepRep for Single Event Visualization

- HepRep is a generic interface for component or client server event displays
- such as
  - WIRED (Java)
  - FRED (C++ OpenGL)





# HepRep Purpose

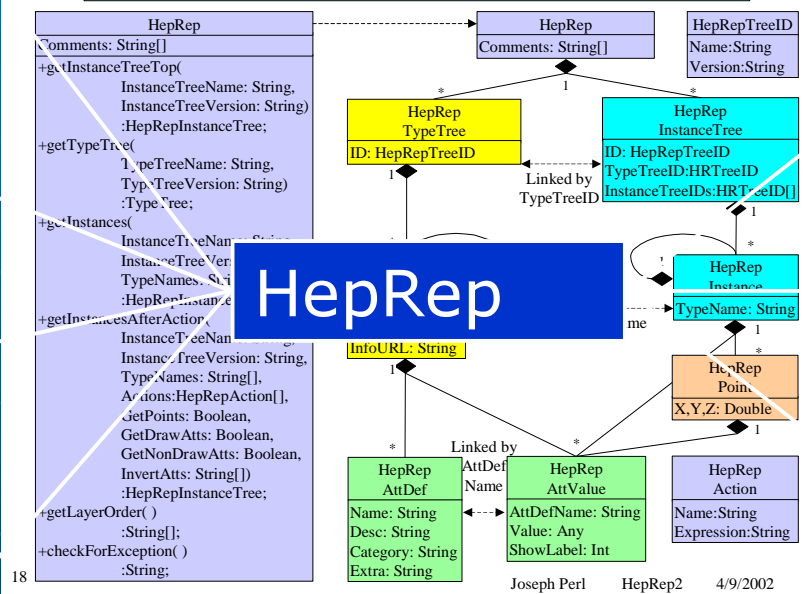
BaBar Server

GLAST Server

LCD Interface

Geant4 Server

HepRep: a Generic Interface Definition for HEP Event Display Representables



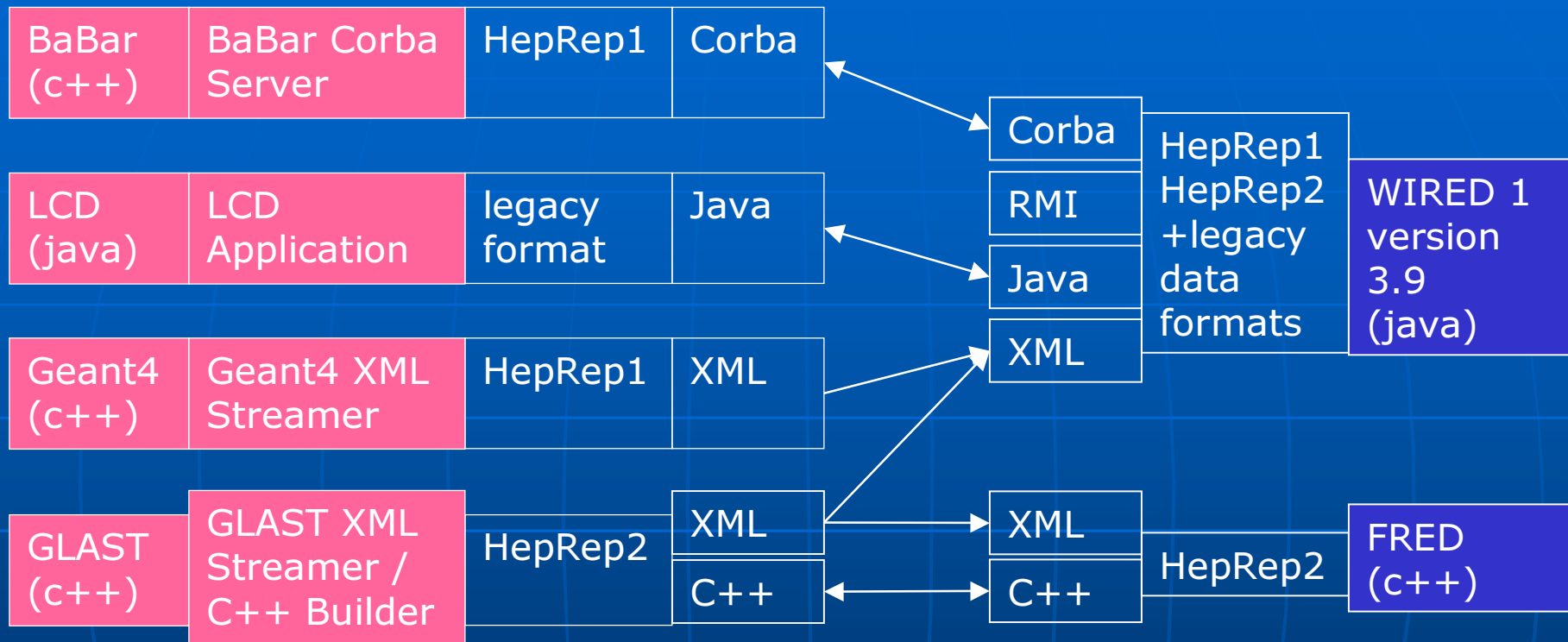
WIRED Client

FRED Client

Other HepRep Clients

The HepRep interface breaks the dependency between any particular experiment's event display server and any particular event display client. The HepRep format is independent of any one particular language or protocol. It can be used from C++ or Java and can be shipped as Corba, RMI, XML, Java or JNI for consumption by WIRED or any other HepRep-enabled event display client.

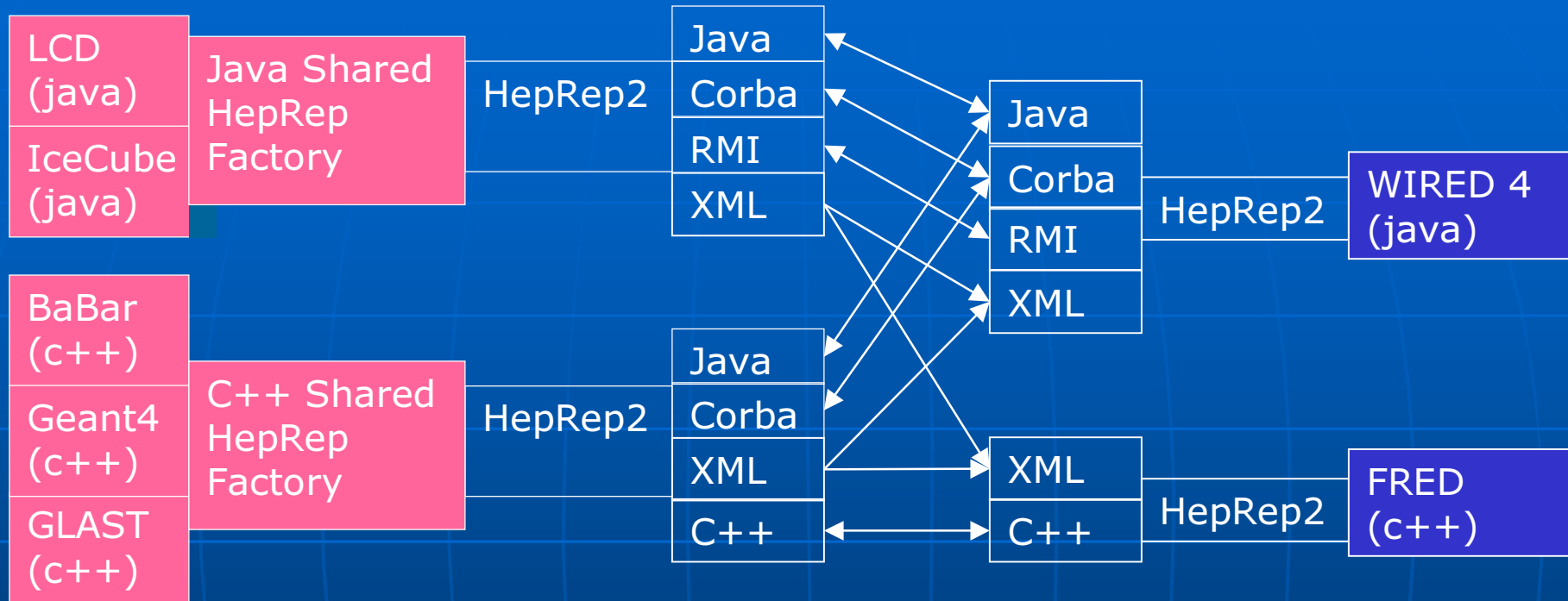
# HepRep Current Use Architecture



While all four SLAC data sources are now using WIRE, they communicate with WIRE in different ways:

- BaBar has its own HepRep1 Corba server, dependent on BaBar code.
- LCD passes WIRE java objects using a legacy data format (pre-HepRep).
- Geant4 has its own HepRep1 XML streamer, dependent on Geant4 code.
- GLAST, being the newest, is leading us towards the right approach.

# HepRep Near-Term Future Architecture



All data sources speak HepRep2 to an abstract HepRep factory. By instantiation of one or another concrete implementation of HepRep:

- a C++ program can change from creating HepRep in C++ memory
- to creating HepRep as an XML streamer (a pure C++ solution with no external library dependencies and no creation of the HepRep in memory)
- to creating HepRep as Corba (depends on Corba libraries)
- or creating HepRep as Java (via Java Native Interface)

# Demo WIRED Reading HepRep Files from the feb 2002 G4 Workshop CD

**Geant4 Workshop Companion CD**  
February 17, 2002  
SLAC

**Selected Item(s)**

Attribute	Value
State	0
LVol	magnetCL
Material	Air
Scid	magneticT
Density	0.001547...
EType	04Tubs
Radius	295161

**Orientation Actions**  
Orientation Toolbar  
Projection  
Mouse Function  
Drawing Options  
Export Graphics...

**View**  
PickRectangle  
PickNode  
PickCircle  
PickPolygon  
PickToMeasure  
When Seen is Selected  
Show Attributes  
Make Viewable in Event Visibility Control  
Parallel Projection  
Scaling  
Translation  
Rotation

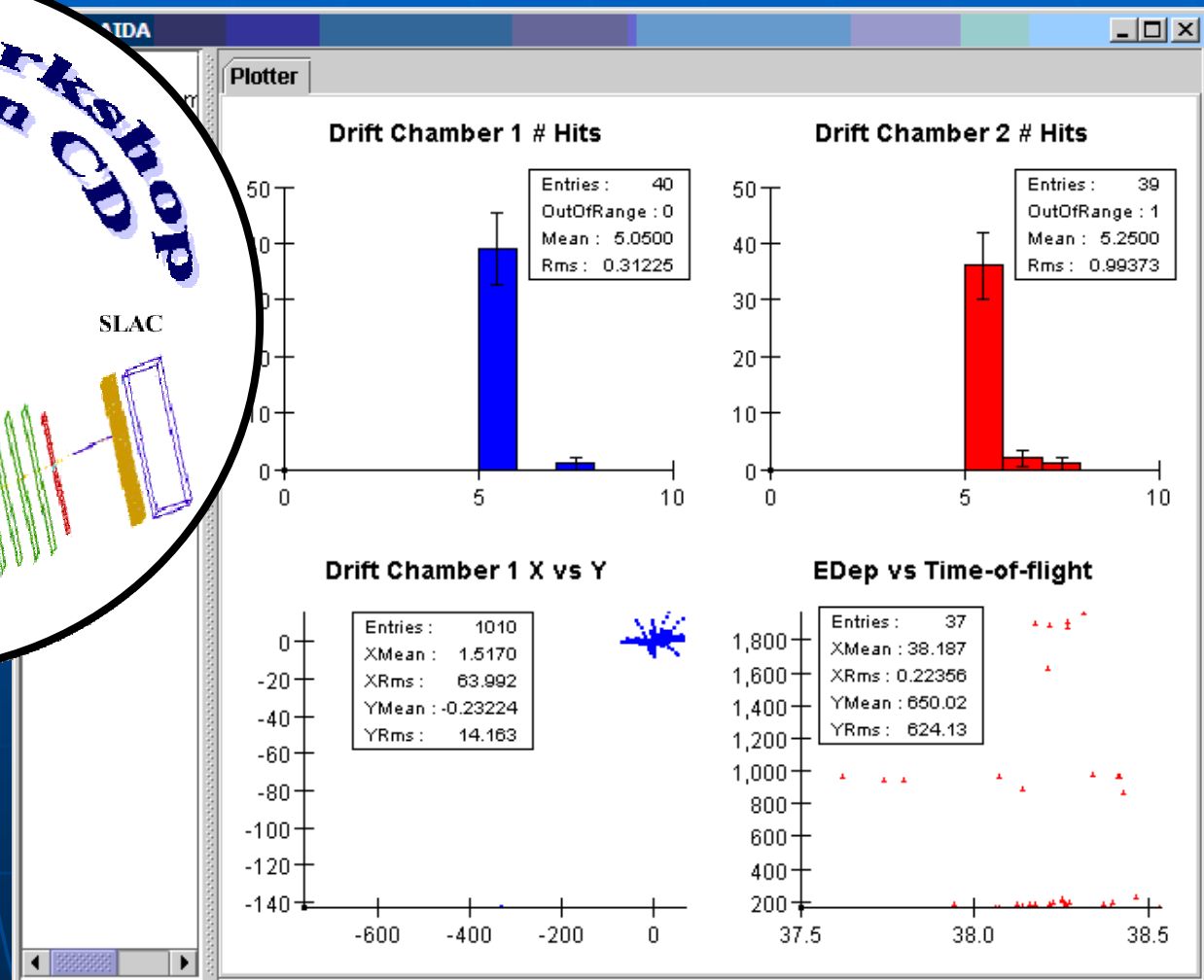
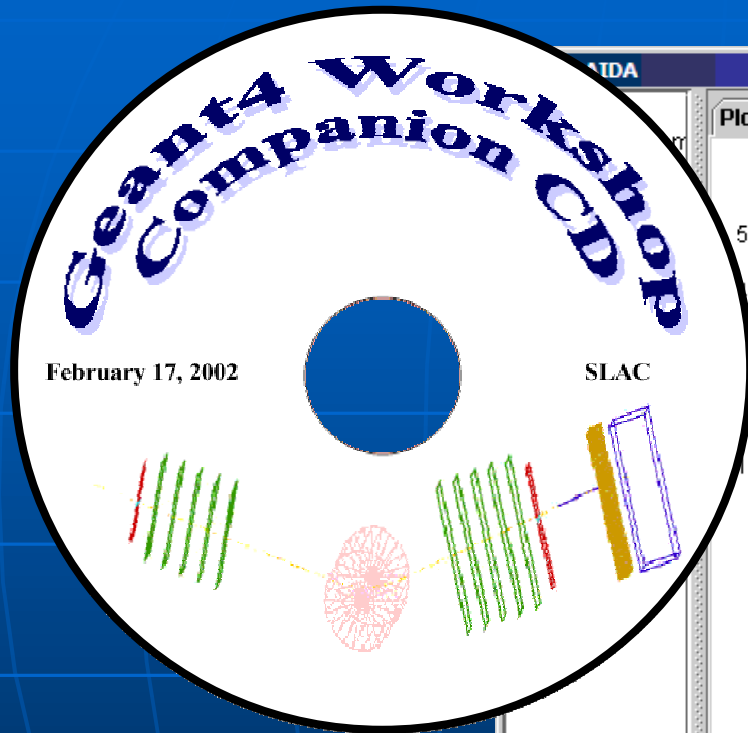
This tutorial was presented at the Geant4 Users Workshop at SLAC in February 2002, but can also be used as a general WIRED tutorial.

An earlier Geant 4 tutorial had users run a basic Geant 4 job contained in Geant4/source/examples/Tutorial/T01. This produced various kinds of output including a graphics file in a format called HepRep (for High Energy Physics REPresentables). This visualization tutorial will show you how to use the WIRED Event Display to view that HepRep file (or an equivalent file that you can access over the web - see details later).

WIRED is a 100 percent pure Java application that can run anywhere that Java version 1.3 or newer is installed. It is currently being used on all versions of Windows (from Windows 95 to the latest), Linux, Unix and Mac OSX. WIRED, and the HepRep graphics protocol, are general purpose tools that can read data from many sources (such as the BaBar collaboration's offline and live data servers and the EGS shower simulation program).

You have already seen the mouse used for picking objects to see their attri...

# Demo JAS Reading AIDA Files from the feb 2002 G4 Workshop CD



# Conclusions

- Component architecture based on generic interfaces has worked well in Geant4.
- All attendees at the Feb 2002 G4 Users Workshop tutorials were successful in using JAS and WIRED to visualize G4 data.
- But these tutorials relied on the user having the pre-built binaries that we produced for that workshop, and the documentation was maintained only on that workshop's web pages.
- These tutorials should be:
  - adapted to the latest G4 version
  - included in the G4 distribution as G4 extended or advanced examples
  - documented in the G4 standard documentation set
  - and perhaps also used as standard G4 test cases



# References

- AIDA: Abstract Interfaces for Data Analysis  
<http://aida.freehep.org>
- JAS: Java Analysis Studio  
<http://jas.freehep.org>
- HepRep: a generic interface definition for HEP event display representables  
<http://heprep.freehep.org>
- WIRED: world wide web interactive remote event display (a HepRep Client)  
<http://www.slac.stanford.edu/BFROOT/www/Computing/Graphics/Wired>
- SLAC HepRep WIRED Work Plan  
<http://www.slac.stanford.edu/~perl/wired>
- Fred: oh no, another event display (a HepRep client)  
<http://www.fisica.uniud.it/~riccardo/research/fred>
- Geant4 Users Workshop at SLAC Feb 2002, Tutorials  
<http://geant4.slac.stanford.edu/UsersWorkshop/tutorial.html>
- A Component Approach to HEP Event Displays  
<http://www.slac.stanford.edu/~perl/component>