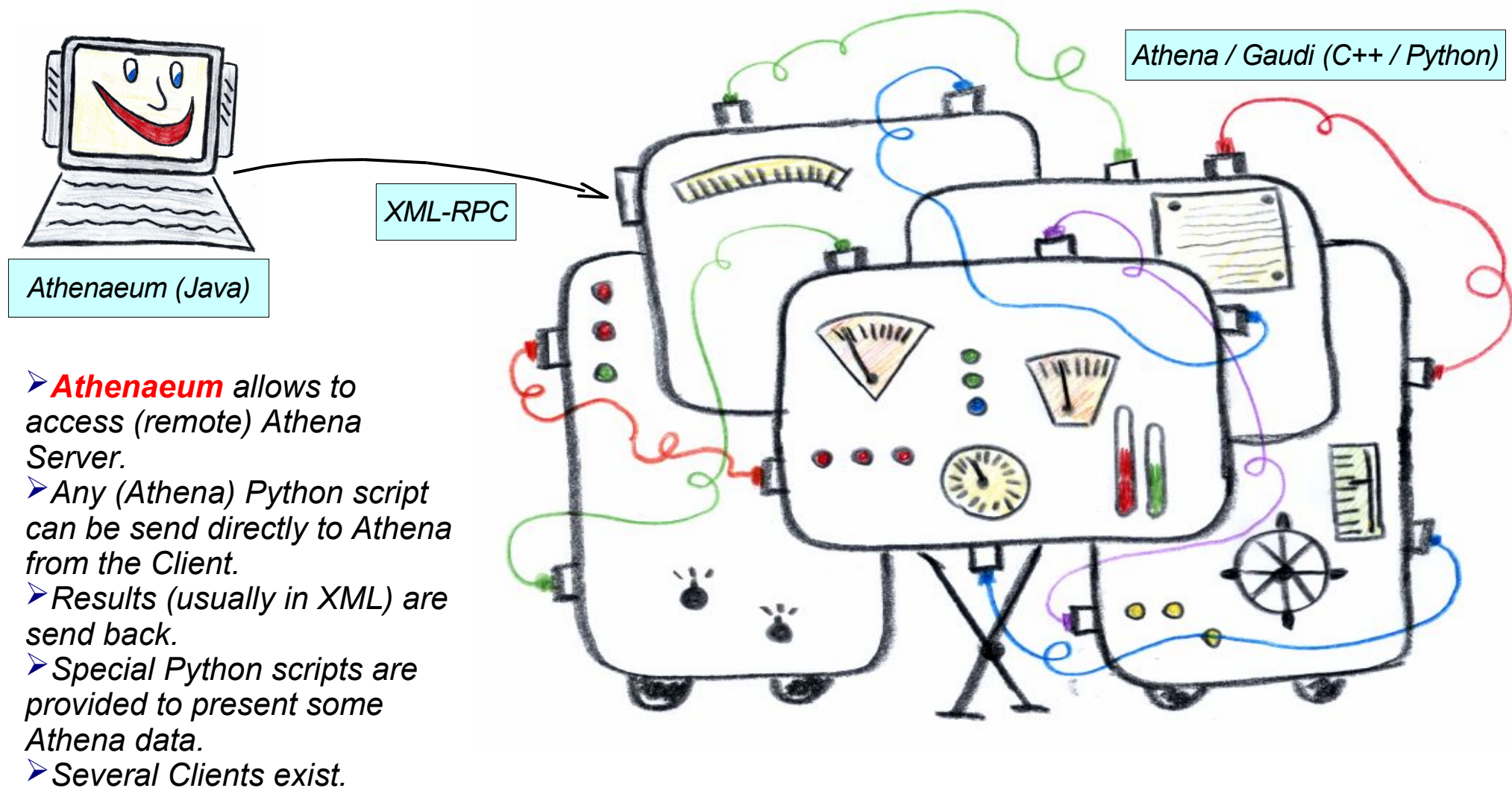




Athenaeum

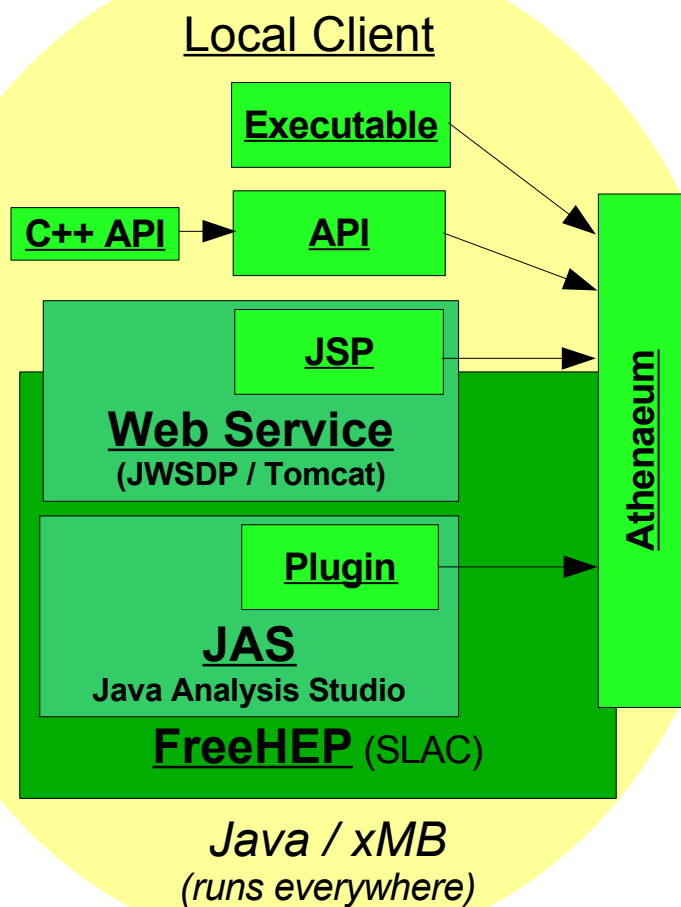
Remote Client to
Atlas Offline Framework



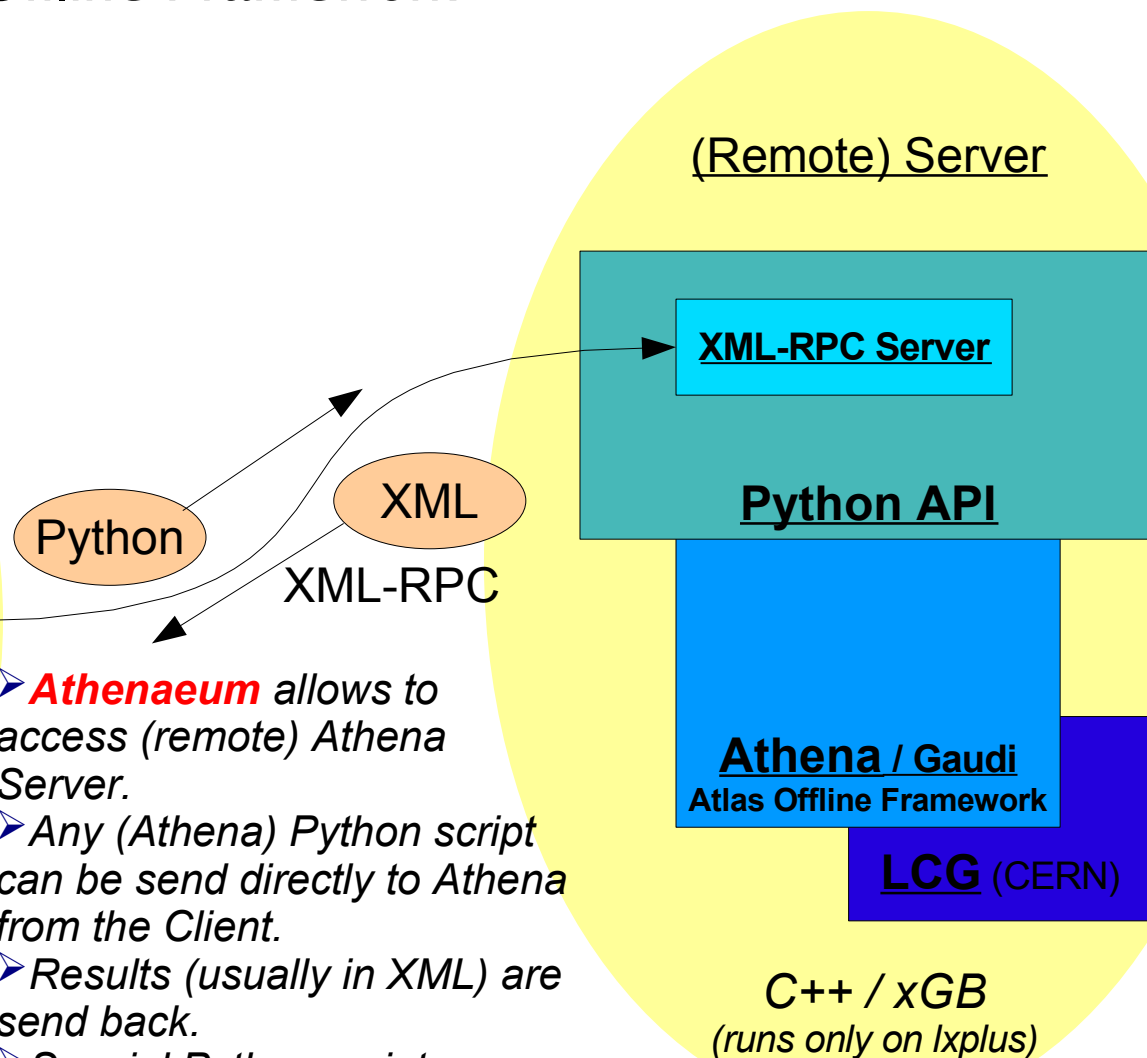


Athenaeum

Remote Client to
Atlas Offline Framework



- **Athenaeum** allows to access (remote) Athena Server.
- Any (Athena) Python script can be send directly to Athena from the Client.
- Results (usually in XML) are send back.
- Special Python scripts are provided to present some Athena data.
- Several Clients exist.





Java Analysis Studio

FREE



GUI

Tree of Objects

Integrated Help with Executable Examples

Java Class

Python/PNuts Script

Python/PNuts Command Line

Graphical/Textual Object Representation

Athenaeum - Athena JAS3 Plugin

- How to Use JAS
- How To Start Athena Python Script
- How To Connect to Athena Python
- How to Work with Local Scripts
- How to Work with Remote Scripts
- How to Loop over Events
- How to Work with Proxies of Athena
- How to Write Proxies of Athena
- Existing Proxies:

Fit.java

```
1 import hep.aida.*;
2 import java.util.Random;
3
4 public class Fit
5 {
6     public static void main(String[] args)
7     {
8         // Create factories
9         IAnalysisFactory analysisFactory = IAnalysisFactory.create();
10        ITreeFactory treeFactory = analysisFactory.createTreeFactory();
11        ITree tree = treeFactory.create();
12        IPlotter plotter = analysisFactory.createPlotterFactory().create("Fit.java Plot");
13        IHistogramFactory histogramFactory = analysisFactory.createHistogramFactory(tree);
14        IFunctionFactory functionFactory = analysisFactory.createFunctionFactory(tree);
15        IFitFactory fitFactory = analysisFactory.createFitFactory();
16
17        IHistogram1D h1 = histogramFactory.createHistogram1D("Histogram 1D", 50, -3, 3);
18
19        Random r = new Random();
20
21        for (int i=0; i<100000; i++) {
22            h1.fill(r.nextGaussian());
23            h1.fill(r.nextDouble()*10-5);
24        }
25
26        IFunction gauss = functionFactory.createFunctionFromScript("gauss", 1, "background+a*exp(-a*(x-mu)**2)");
27        gauss.setParameter("a", h1.maxBinHeight());
28        gauss.setParameter("mu", h1.mean());
29    }
30 }
```

Python/PNuts Script

```
>>> print "Hello"
Hello
>>>
```

Graphical/Textual Object Representation

h2d

Entries: 9999

Mean: -8.9443E-4

Rms: 1.0160

OutOfRange: 1

JAS is a GUI based on FreeHEP library.

FreeHEP is Java equivalent of CERNLIB, Root, OpenScientist,...

Most Functionality implemented by Plugins.

They can be loaded dynamically (over network).

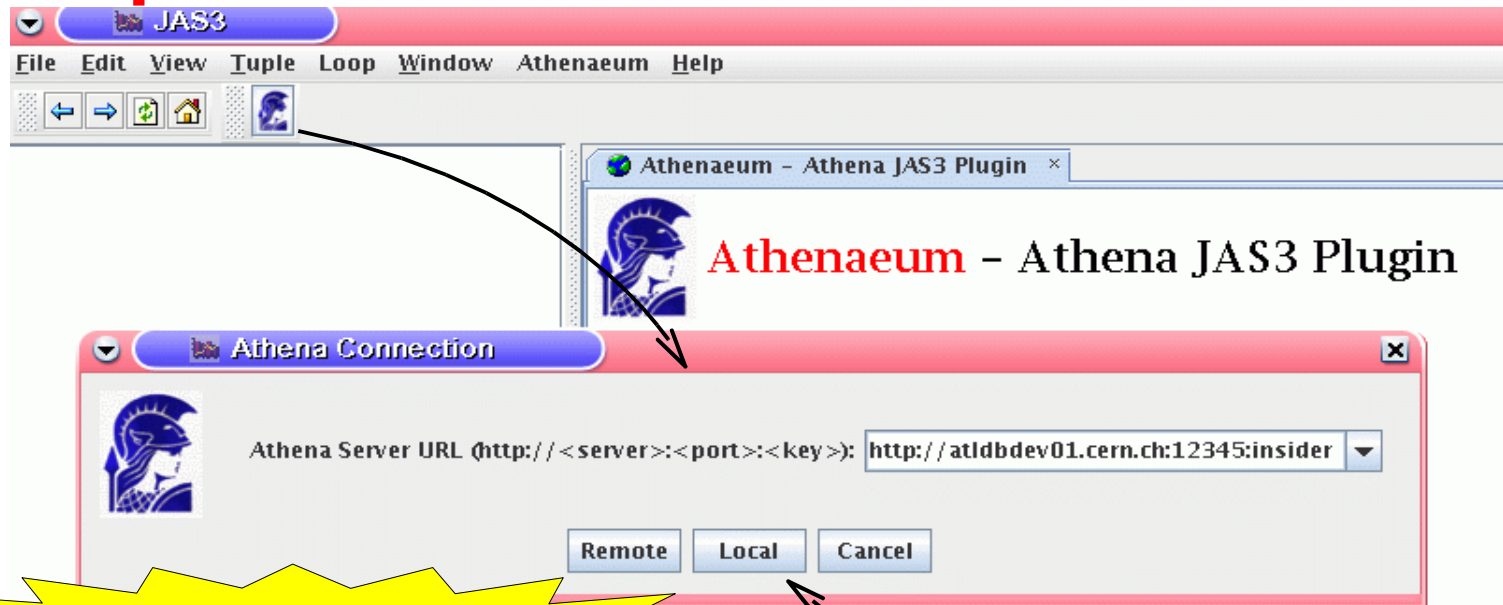
9.59/11.8MB

see <http://jas.freehep.org/jas3> for details



Atlas Offline Framework (C++ / Python)

Open Connection to Athena



*On Client
(Any platform with JAS + Athenaeum Plugin)*

*Local mode runs without Server,
it can read XML files
which would otherwise be
obtained from the server*

```
$ athena.py -i -s jobOptions.py
```

```
.....  
XML-RPC server 'atldbdev01.cern.ch:48966' created  
method 'process()' registered  
Waiting for requests...
```

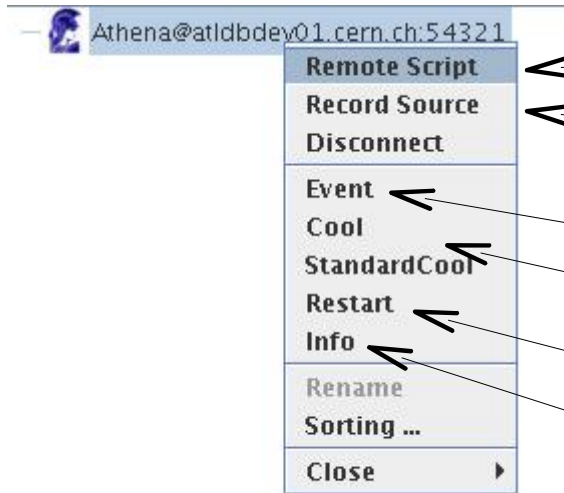
*On Server
(Linux with Athena)*

```
.....  
execfile ("InteractiveServer.py")
```

*Server script written
by Atlantis team
(<http://cern.ch/atlantis>)*



Interact with Athena



- Execute a Python script on Athena Server, get results back
- Steer Athena Event Loop from JAS
- Access Event data
- Access Cool data
- Restart Athena server
- Get Information about Athena Server environment (loaded dictionaries, paths,...)

*Registered **Proxies** are implementing concrete handling of connection to specific (Athena) functionality.*



Execute Python on Athena

Script to be executed on remote Athena

```
Athena@lxplus003.cern.ch:48966
1 print self
```

Script Result

```
Result_1@lxplus003.cern.ch:48966
1 <__main__.InteractiveServer instance at 0xb720830c>
2
3 SUCCESS
```

Output Console

```
Connecting to http://lxplus003.cern.ch:48966
On http://lxplus003.cern.ch:48966 executing:
-----
print self
Result:
-----
<__main__.InteractiveServer instance at 0xb720830c>
```

classpath:/org/freehep/jas/web/relnotes.html 4.65/6.13MB

*User can mix Python running within JAS
and Python running in a (remote) Athena.
Athena Python scripts could be moved to JAS.*



Steer Athena Event Loop

The screenshot displays the JAS3 IDE interface. The top menu bar includes File, Edit, View, Tuple, Loop, Window, Athenaem, and Help. The toolbar contains navigation and execution icons. The left sidebar shows a project tree with 'Athena@lxplus003.cern.ch:48966' and 'DataSets'. The main editor window shows the 'EventLoop.java' file with the following code:

```
1 import org.freehep.record.loop.event.RecordAdapter;
2 import org.freehep.record.loop.event.RecordSuppliedEvent;
3
4 import net.hep.atlas.Core.Athenaem.JAS3Plugin.AthenaClient;
5
6 public class EventLoop extends RecordAdapter {
7
8     public void recordSupplied(RecordSuppliedEvent event) {
9         AthenaClient athena = (AthenaClient)event.getRecord();
10        try{
11            System.out.println(athena.execute("print self"));
12        }
13        catch (Exception e) {
14            System.err.println(e);
15            e.printStackTrace();
16        }
17    }
18
19 }
20
```

Annotations and callouts:

- Next Event,...**: Points to the 'Next' button in the toolbar.
- Athena interpreted as a set of Records (Events)**: Points to the 'DataSets' folder in the sidebar.
- Python script executed on each Event Results analyzed locally**: Points to the `athena.execute("print self")` line in the code.
- Output Console**: Points to the console output area at the bottom.

Console Output:

```
Connecting to http://lxplus003.cern.ch:48966
On http://lxplus003.cern.ch:48966 executing:
-----
theApp.initialize()
```

Bottom status bar: classpath:/net/hep/atlas/Core/Athenaem/JAS3Plugin/doc-files/EventLoop.java 6.41/8.77MB



Remote Proxy

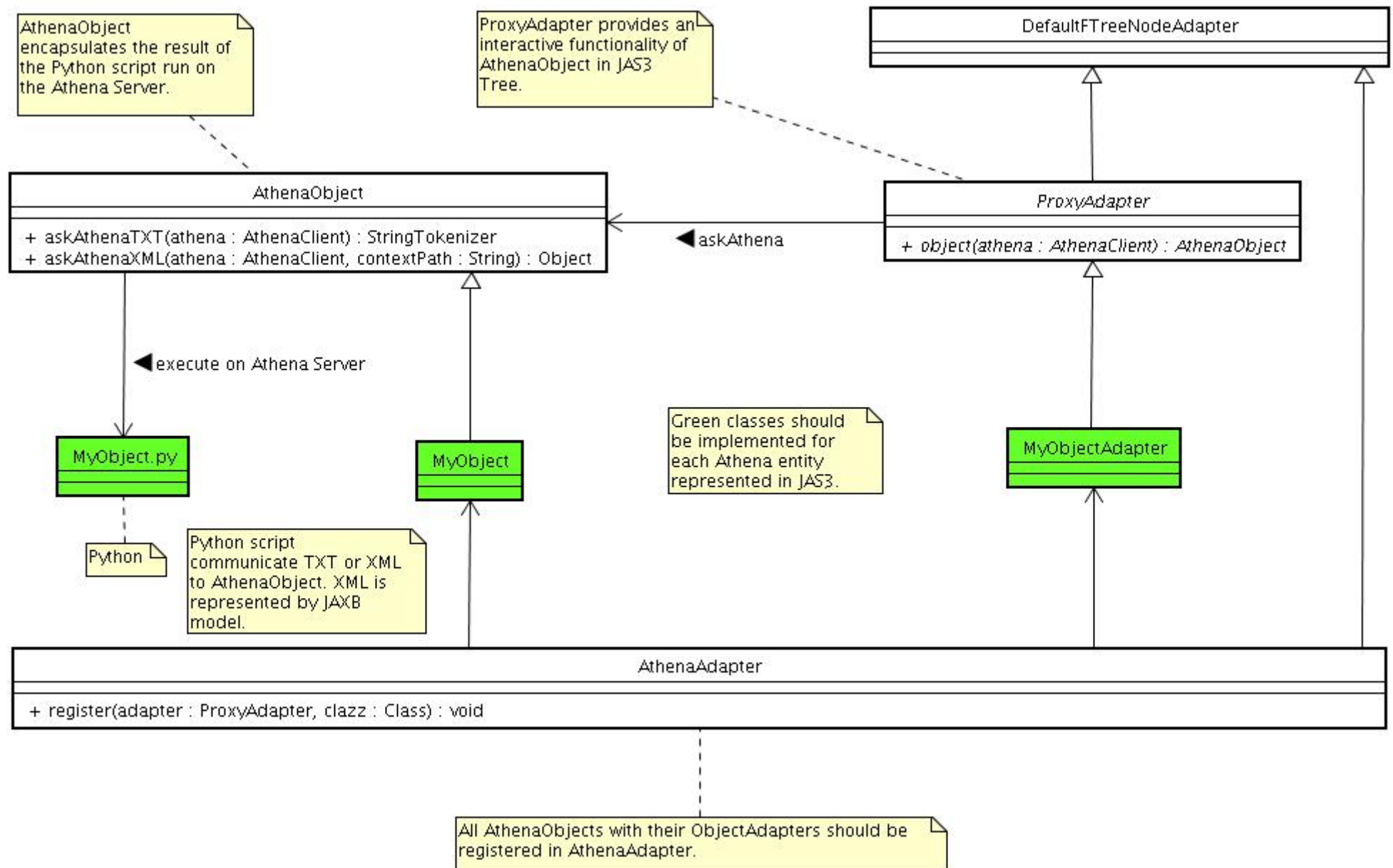
*Registered **Proxies** are implementing concrete handling of connection to specific (Athena) functionality. They are implemented by:*

- Athena Python script to extract data from Athena*
- JAS wrapper to present/handle data inside JAS*
- XML schema to describe data*

When implementing pre-defined interfaces from Athenaeum, those Proxies will make themselves automatically available inside JAS system in an organic way.

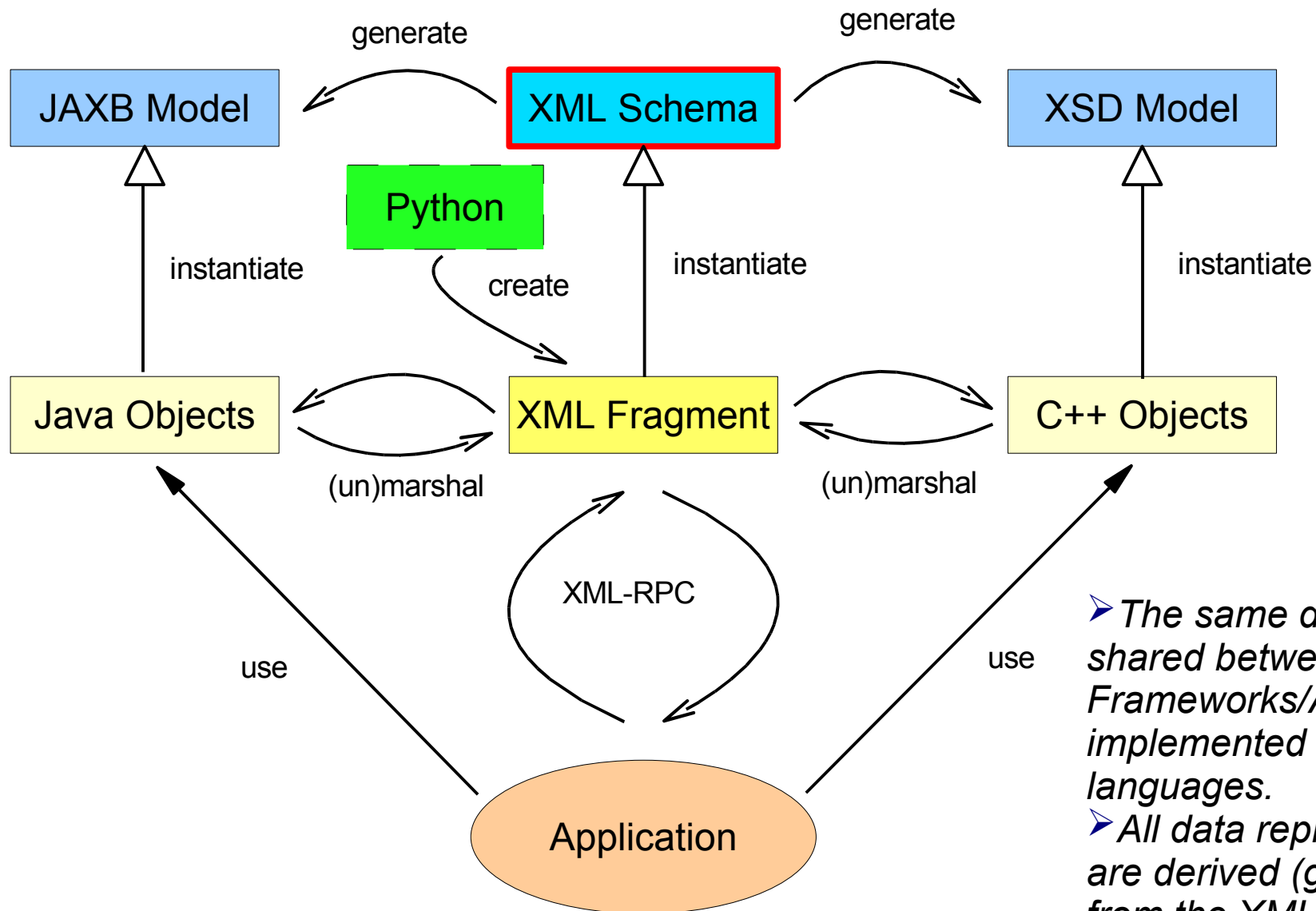


Construction of Proxy





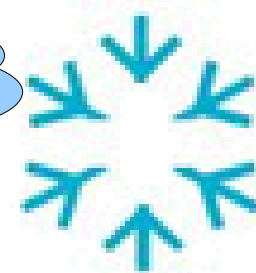
XML Schema Representations



- The same data are shared between different Frameworks/Applications implemented in different languages.
- All data representations are derived (generated) from the XML Schema.



LCG Conditions Database
(C++ / Python / SQL)



Interact with Cool

Cool Connection

Server: oracle://INTR
Database: LARMC130
Schema: ATLAS_COOL_LAR
User: ATLAS_COOL_READER
Password: COOLRED4VAL
Tag:
Channels:
Since: 1/1/00 12:00 AM
Until: 6/22/06 6:19 PM
StartRun: 0
StartEvent: 0
StopRun: 2147483647
StopEvent: 4294967295
Payload: false

Full Summary Cancel

JAS + Athenaueum
Client

Athena/PyCool
Server

Cool DB
Server

- Open connection to Cool DB
- Interpret data (as AIDA NTuples)
- Show data as HTML
- Show data as XML
- Show data in a Tree View
- Show Python script used to get data
- Show XSLT stylesheet used to create HTML

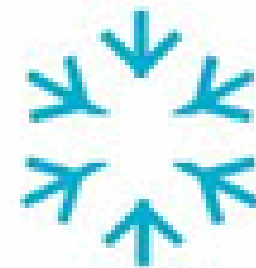
Athena@lxplus067.cern.ch:2389

Cool@oracle:ATLAS_COOLPROD,ATLAS_COOL_INDET]_0

Convert into NTuple
Show as HTML
Show as XML
Show as Tree
Edit Script
Edit XSLT
Rename
Close



Cool XML

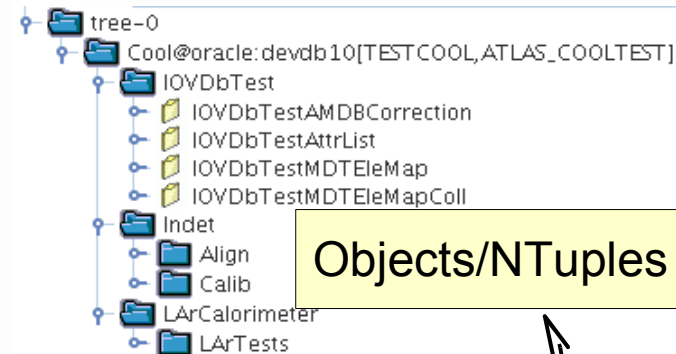


/IOVDbTest/IOVDbTestAttrList[1326]

Inserted: 24/10/2005 at 4:58:13
FOLDER_IOVTABlename: TESTCOOL_F1326_IOVS
FOLDER_TAGTABlename: TESTCOOL_F1326_TAGS
FOLDER_IOV2TAGTABlename: TESTCOOL_F1326_IOV2TAG
Channels: 0,
TypeName: AthenaAttributeList
TimeStamp: run-event
Symlinks:
ServiceType: 71
Clid: 40774348

HTML View

period	object	channel	insert	xPosition [float]	id [int]	name [string]
run:0 event:0 run:3 event:3	8	0	24/10/2005 at 5:0:17	25.0	7	TestAttrList
run:3 event:3 run:4 event:3	14	0	24/10/2005 at 5:0:41	25.0	7	TestAttrList
run:4 event:3 run:2147483647 event:4294967295	13	0	24/10/2005 at 5:0:41	25.0	7	TestAttrList



Objects/NTuples

Java

/IOVDbTest/IOVDbTestAttrListColl[1327]

Inserted: 24/10/2005 at 4:58:46
FOLDER_IOVTABlename: TESTCOOL_F1327_IOVS
FOLDER_TAGTABlename: TESTCOOL_F1327_TAGS
FOLDER_IOV2TAGTABlename: TESTCOOL_F1327_IOV2TAG
Channels: 16, 26, 36, 46, 56,
TypeName: CondAttrListCollection
TimeStamp: run-event
Symlinks:
ServiceType: 71
Clid: 1238547719

XSLT

```
<folder name='/IOVDbTest/IOVDbTestMDTEleMapColl'
  id='1331' day='24' month='10' year='2005' hour='5' minute='2' second='36' >
  <attributes>
    <attribute name='FOLDER_IOVTABlename' value='TESTCOOL_F1331_IOVS'/>
    <attribute name='FOLDER_TAGTABlename' value='TESTCOOL_F1331_TAGS'/>
    <attribute name='FOLDER_IOV2TAGTABlename' value='TESTCOOL_F1331_IOV2TAG'/>
  </attributes>
  <channels>
    <channel>0</channel>
  </channels>
  <description>
    <timeStamp>run-event</timeStamp>
    <addrHeader><address_header service_type="71" clid="155887251" /></addrHeader>
    <typeName>IOVDbTestMDTEleMapColl</typeName>
  </description>
  <signature>
    <item name='PoolRef' type=' string'/>
  </signature>
  <payload since='run:0 event:0' until='run:2147483647 event:4294967295'
    object='1' channel='0' day='24' month='10' year='2005' hour='5' minute='3' second='6' >
    <entry name='PoolRef' value='[...]'>
    </entry>
  </payload>
</folder>
```

XML View

XML Schema

Python



Work with Cool (1)



- Data can be represented as
 - XML
 - Objects
 - Tree
 - (AIDA) NTuples
 - HTML
- and accessed
 - via GUI
 - using scripting interface (Java, Python, Pnuts)
 - using API (Java, Python)

The screenshot shows the JAS3 Cool Proxy for Athena interface. The main window displays a tree view of data. The tree structure is as follows:

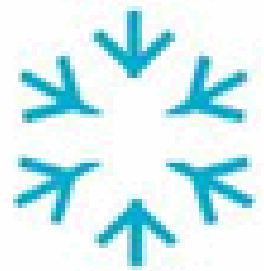
- athenaeum
 - server="lxplus058.cern.ch:14420"
 - cool
 - dbname="TESTCOOL"
 - password="pwd4cool"
 - schema="ATLAS_COOLTEST"
 - server="oracle://devdb10"
 - user="ATLAS_COOLTEST"
 - #comment
 - #comment
 - folder
 - day="12"
 - hour="15"
 - id="2707"
 - minute="58"
 - month="0"
 - name="/IOVDbTest/IOVDbTestAMDBCo"
 - second="50"
 - year="2006"
 - attributes
 - channels
 - description
 - signature
 - payload
 - channel="0"
 - day="12"
 - hour="16"
 - minute="1"
 - month="0"
 - object="8"
 - second="40"
 - since="1970/01/01 Thu 00:00:00"
 - until="1970/01/01 Thu 00:00:35"
 - year="2006"
 - entry
 - #comment

The right pane shows the details of the selected node, which is the 'payload' folder. The details include:

- until="1970/01/01 Thu 00:00:35"



Work with Cool (2)

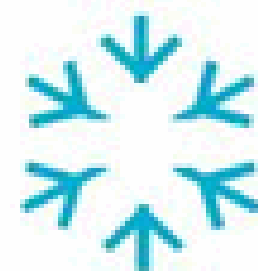


A screenshot of the JAS3 software interface showing a project tree on the left and a data table on the right. The project tree shows a hierarchy starting with 'Cool' and 'tree-0', leading to 'IOVDbTestAttrList' which lists attributes like 'since', 'until', 'object', etc. The data table on the right shows three rows of data for 'run:0 e...', 'run:3 e...', and 'run:4 e...', with columns for time, object, channel, day, month, year, hour, minute, second, xPosition, id, and name. Arrows point from the table to the list of attributes in the tree, and from the list of attributes to a bulleted list of representation and access methods.

Cool Proxy for Athenaem **Cool Result** **IOVDbTestAttrList**

since	until	obj...	cha...	day	month	year	hour	minute	second	xPosition	id	name
run:0 e...	run:3 e...	8	0	31	9	20...	18	46	28	25.0	7	TestAttrList
run:3 e...	run:4 e...	14	0	31	9	20...	18	46	51	25.0	7	TestAttrList
run:4 e...	run:21...	13	0	31	9	20...	18	46	51	25.0	7	TestAttrList

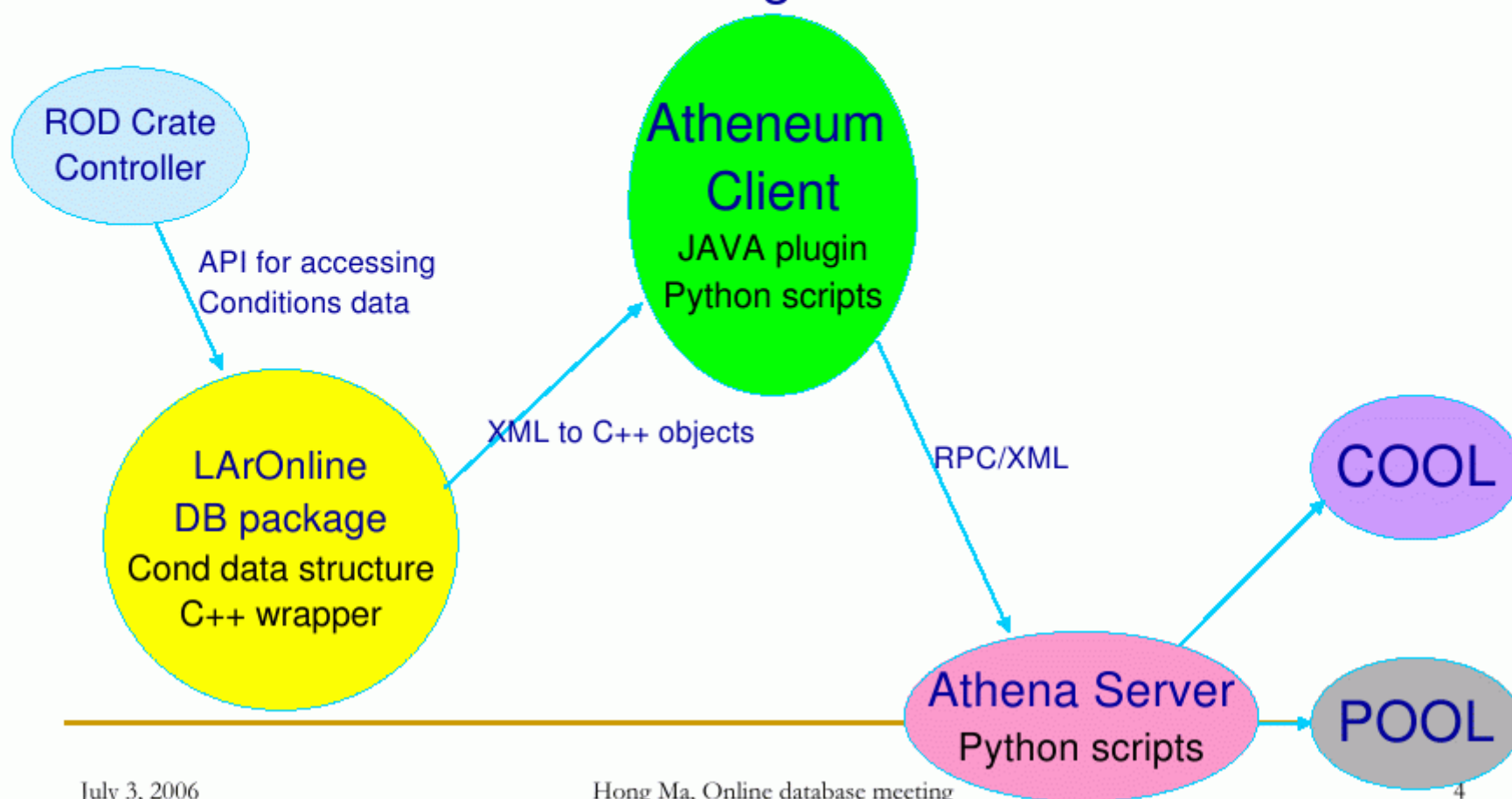
- Data can be represented as
 - XML
 - Objects
 - Tree
 - (AIDA) NTuples
 - HTML
- and accessed
 - via GUI
 - using scripting interface (Java, Python, Pnuts)
 - using API (Java, Python)



LAr Cool Online

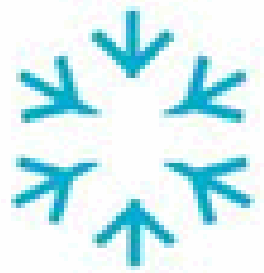
Client/server for LAr online COOL/POOL

Hucheng CHEN

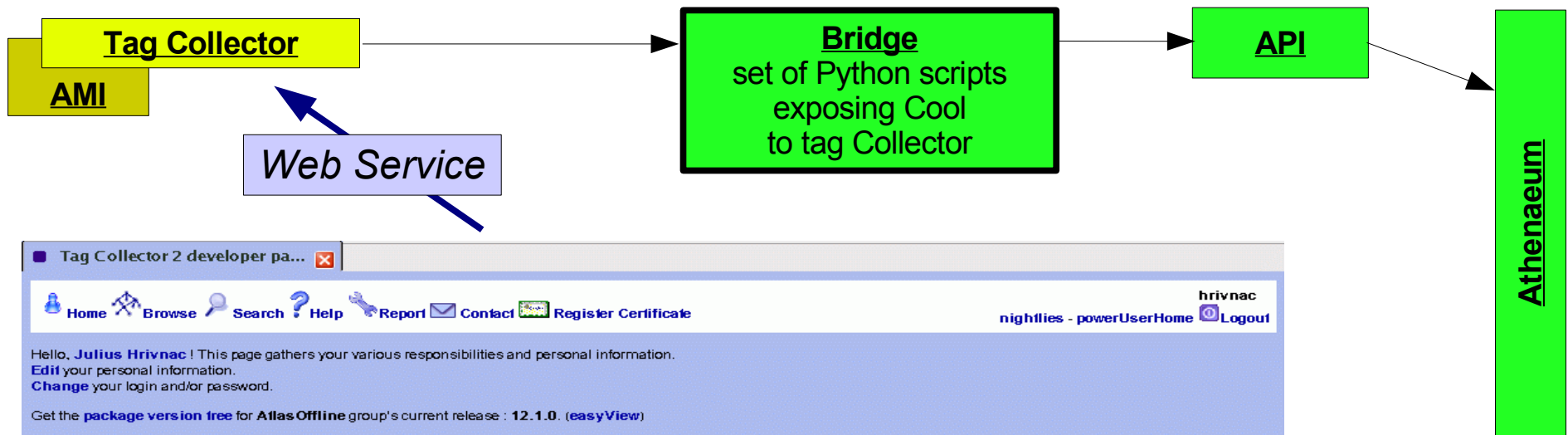




Cool Tag Collector



- Using Tag Collector (Web Service) to handle Cool data
- Work in progress (with Grenoble)



Tag Collector 2 developer pa... [X]

Home Browse Search ? Help Report Contact Register Certificate

hrivnac
nighflies - powerUserHome Logout

Hello, **Julius Hrivnac** ! This page gathers your various responsibilities and personal information.
[Edit](#) your personal information.
[Change](#) your login and/or password.

Get the **package version free** for **AtlasOffline** group's current release : **12.1.0.** ([easyView](#))
Get the **package version free** for **AtlasOffline** group's last validated release : **12.0.0.** ([easyView](#))
Get the **release free** for group **AtlasOffline** and access easy view for all releases.

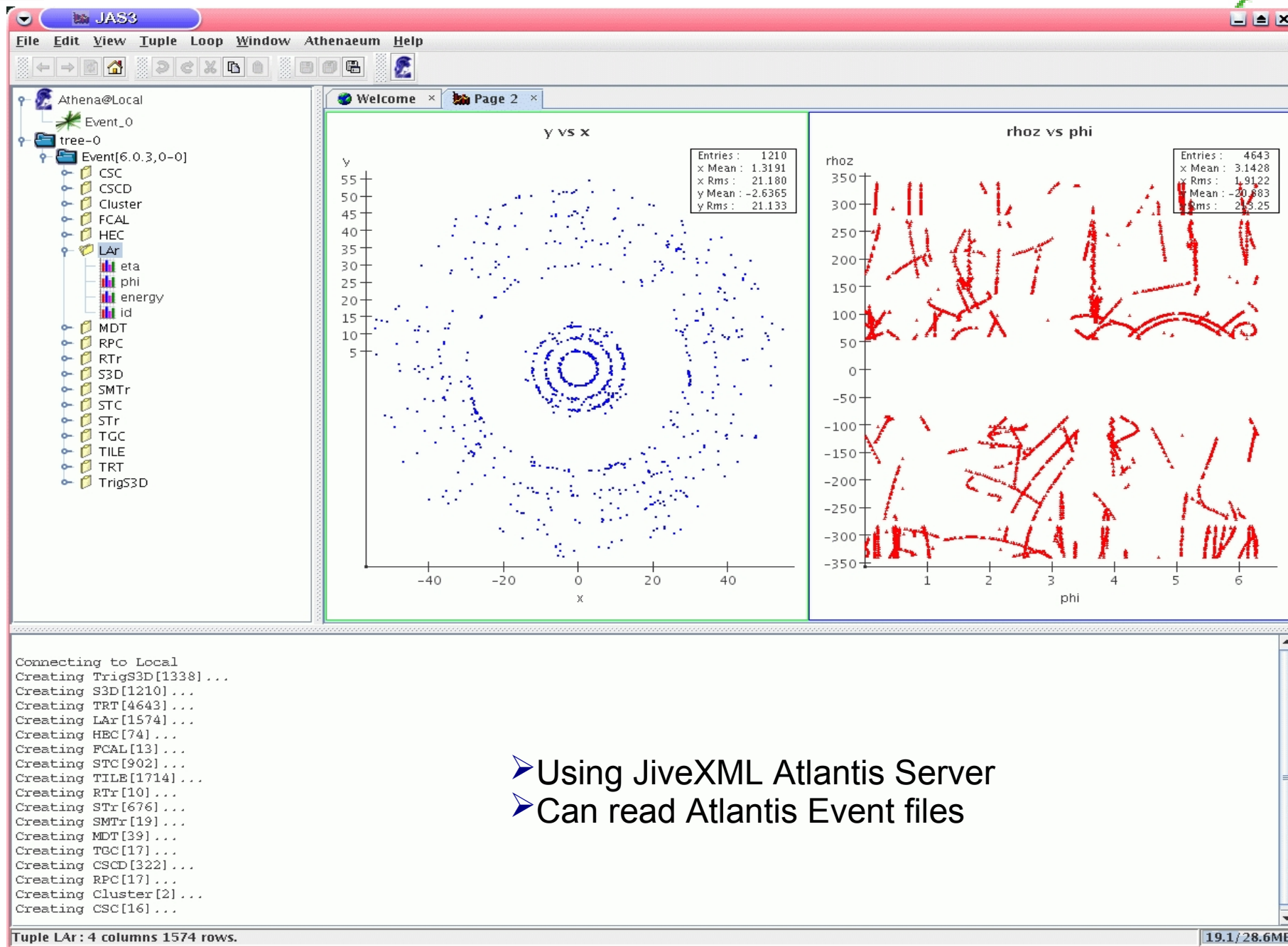
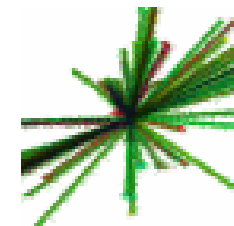
Group filter:

Packages

Package name	Repository name	Role	Command shortcuts
Ant (Show active versions)	AtlasOfflineRepository	tcPackageManager	AtlasCore/2.2.0/Ant-00-00-26 <input type="button" value="Update Tag"/> Actions on sub packages
Axis (Show active versions)	AtlasOfflineRepository	tcPackageManager	AtlasCore/2.2.0/Axis-00-00-06 <input type="button" value="Update Tag"/> Actions on sub packages
CERNJavaInstallation (Show active versions)	AtlasOfflineRepository	tcPackageManager	AtlasCore/2.2.0/CERNJavaInstallation-01-00-15 <input type="button" value="Update Tag"/> Actions on sub packages



Work with Events



- Using JiveXML Atlantis Server
- Can read Atlantis Event files



JSP Web Service

**Athenaeum JSP**
Athena Server URL: :

[Run Script on Server](#)
[Get Server Info](#)
[Browse Cool DB \(connect to any database\)](#)
[Browse Cool DB \(choose from standard databases\)](#)
[Get Help](#)

**Browse Cool DB**
 :
Server:
DB Name:
Schema:
User:
Password:

Athenaeum@atldbdev01.cern.ch:12345
Cool@oracle://devdb10[TESTCOOL,ATLAS_COOLTEST]
/IOVDbTest/IOVDbTestAMDBCorrection[4173]

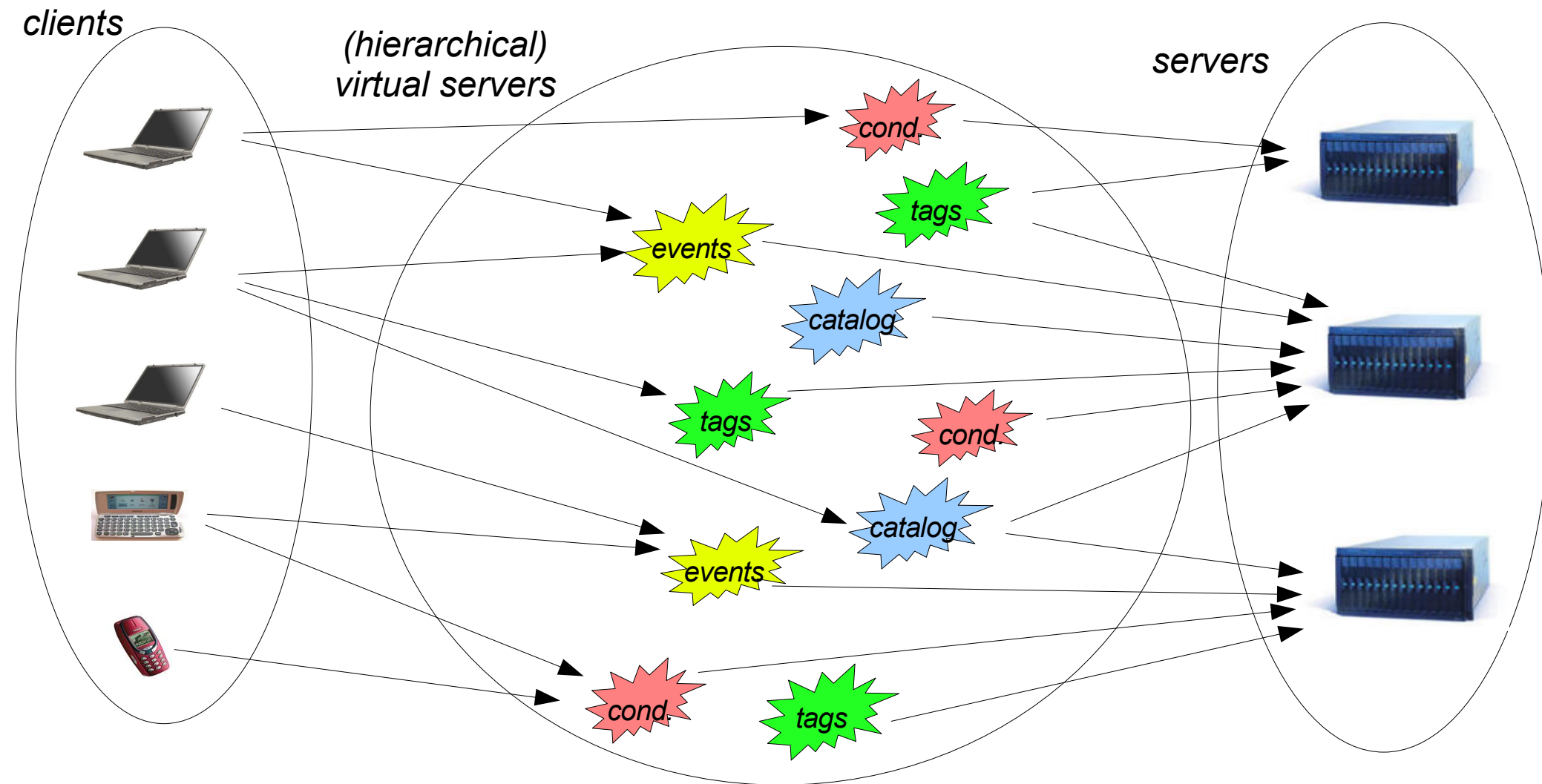
Inserted:	4/3/2006 4:43:51
FOLDER_IOVTABLENAME:	TESTCOOL_F4173_IOVS
FOLDER_TAGTABLENAME:	TESTCOOL_F4173_TAGS
FOLDER_IOV2TAGTABLENAME:	TESTCOOL_F4173_IOV2TAG
Type Name:	IOVDbTestAmdbCorrection
Key:	
TimeStamp:	time
Symlinks:	
ServiceType:	71
Clid:	132798543

period	object	channel	insert	PoolRef [string]
------------------------	------------------------	-------------------------	------------------------	------------------------------------

- More easy to use than JAS Plugin, but less functionality available
- Packaged as a self-consistent WAR-file
- Can be deployed in any standard Web Container (usually Tomcat)
- Currently deployed as <http://cern.ch/Athenaeum>



Distributed Interactive Environment Architecture Project



- Only user code + access layer in clients
- Data access and standard processing in servers
- Orchestration and optimization in virtual servers
- Passed data described by common (XML) Schema

- **Athenaeum**
- SQLTuple/ColMan
- Sequoia



Architecture Advantages

- Light local client
 - Running on any platform, any release
 - Fully interactive GUI, scripting and API in several languages
 - Easily extensible by modular plugins
- Server on a powerful machine, close to data, replicated and hierarchised when useful
- Standard communication protocols
 - XML-RPC for the Control Flow and small data
 - Eventually performant protocols (JDBC, xrootd,...) for big data



Problems

- PyAthena (Python API to Athena)
 - **Incomplete** (only a subset of C++ API is available via Python)
 - **Undocumented** (C++ Doxygen is not enough for documentation of its Python API; it is not easy to guess the meaning of weakly-typed methods; code fragments on Web/Wiky are often out-of-date)
 - **Unstable** (too many things change too often)
 - The **C++** Framework is **still there**, it just hidden (its problems will pop up from time to time)
- Data
 - **No abstract data definition** is available, the actual data model is hidden very deep in the C++ header files forest
 - Athenaeum **XSD Schema** has been written for data passed around; XML, Java, Python and C++ incarnations can be created from them



To Do Next

- Generalization for other Monolithic Frameworks
 - there is nothing special about Athena/Gaudi, any Framework with functional XML-RPC server would work fine
- Lazy & Compressed data transport (to speed up)
 - XML-aware compression, MPEG-7 compression, binary XML,... can give size down to about 2x compressed Root files size of the same data
- More Proxies (Analysis objects, Generic StoreGate access, ...)
- Athena (remotely) startable from Athenaeum (so that user does not have to start the server herself)
- Deployment of a network of hierarchical *Athena Servers*



How To Start

➤ Within CERN AFS:

- `./afs/cern.ch/sw/java/share/bin/setjdk sun 1.5.0_02`
- `/afs/cern.ch/atlas/offline/external/JAS/jas3/jas3`

➤ Elsewhere (any platform):

- Get Java 1.5
- Get JAS from <http://jas.freehep.org/jas3> (Linux, MS, MacOSX,...)
- Set Plugin Server (View - Preferences...)
- Get Plugin (View - Plugin Manager...)

➤ On WWW:

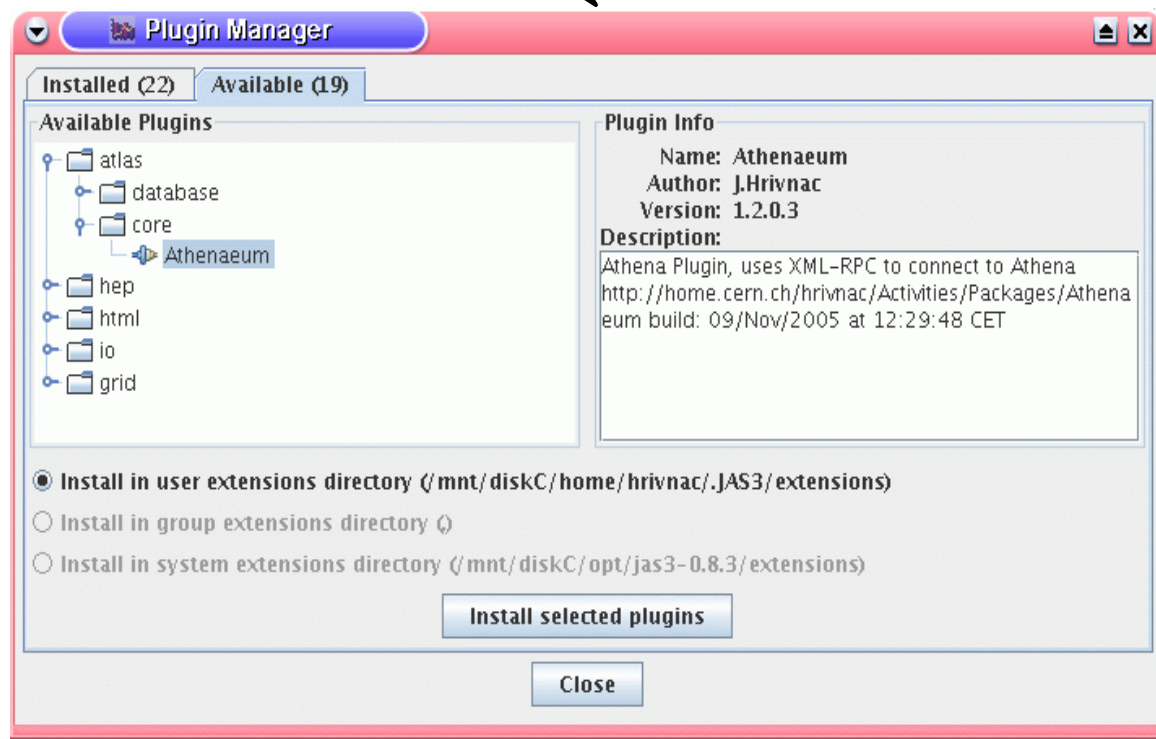
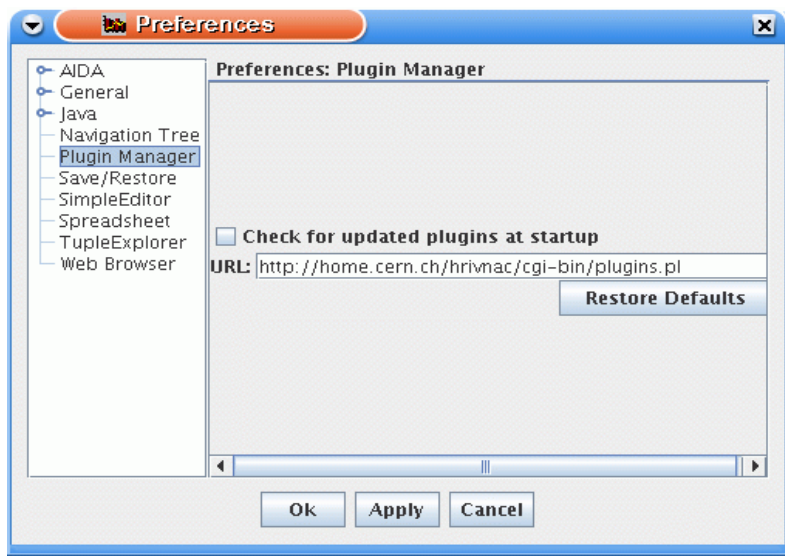
- <http://cern.ch/Athenaeum>

➤ Running Servers:

- <http://atldbdev01.cern.ch:12345>
- <http://atldbdev01.cern.ch:54321>

➤ Personal Server:

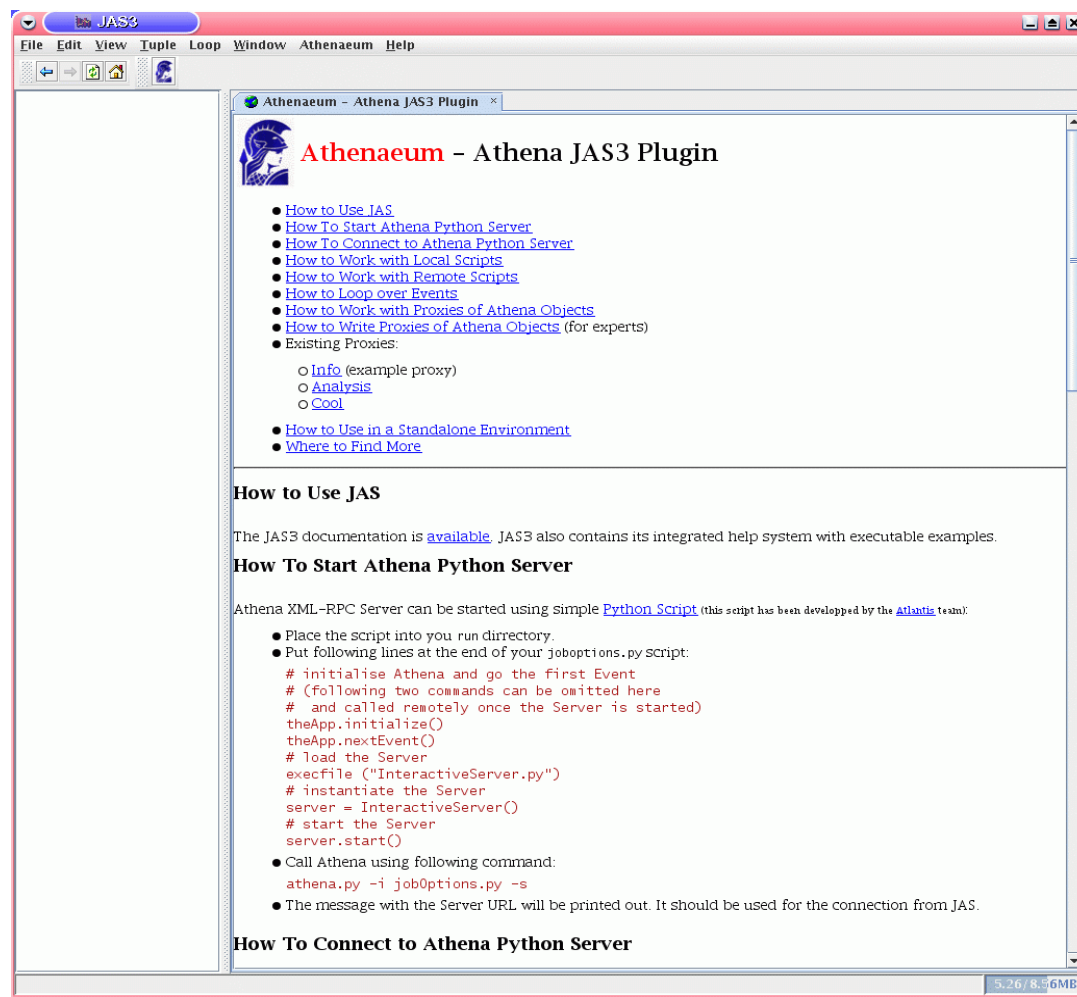
- `offline/Database/AthenaeumServer`





Help

- <http://home.cern.ch/hrivnac/Activities/Packages/Athenaeum>
- <https://uimon.cern.ch/twiki/bin/view/Atlas/HowToUseJAS>
- <http://cern.ch/Athenaeum>
- JAS integrated Help (with executable examples)



*Athenaeum is often updated,
some snapshots in this presentation
may not correspond
to the actual Athenaeum version*