



Athenaeum



Une interface **JAS** au framework **Atlas**

Local Client

Athenaeum Plugin

JAS

Java Analysis Studio

FreeHEP (SLAC)

Java / xMB
(runs everywhere)

(Remote) Server

XML-RPC Server

Python API

Athena / Gaudi
Atlas Offline Framework

LCG (CERN)

C++ / xGB
(runs only on lxplus)

XML

Python

XML-RPC

- **Athenaeum** allows to access (remote) Athena from (local) JAS.
- Any (Athena) Python script can be send directly from JAS.
- Results (usually in XML) are send back and can be processed within JAS.
- Special Python scripts are provided to present Athena data (Cool, etc.) within JAS.

J.Hrivnac
DevDur, Dec'05



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

JAS is a GUI based on FreeHEP library. FreeHEP is Java equivalent of CERNLIB, Root, OpenScientist,...

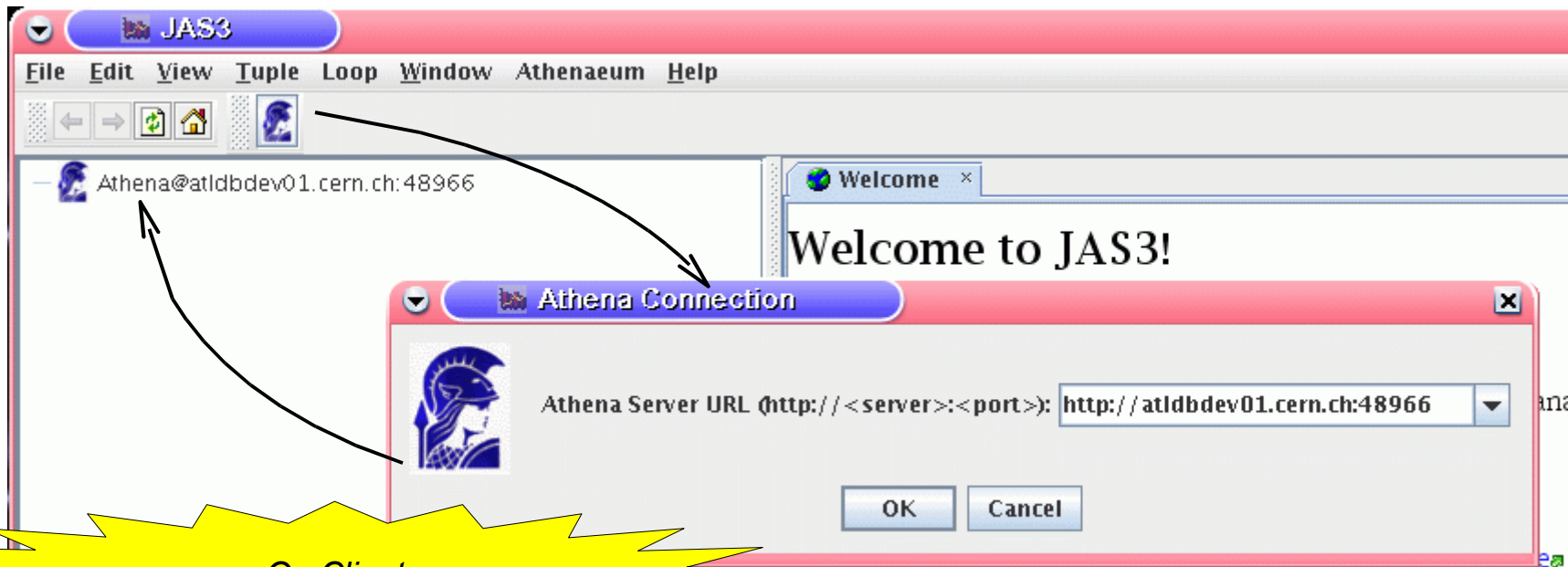
Most Functionality implemented by Plugins. They can be loaded dynamically (over network).

9.59/11.8MB

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



Open Connection to Athena



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

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

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

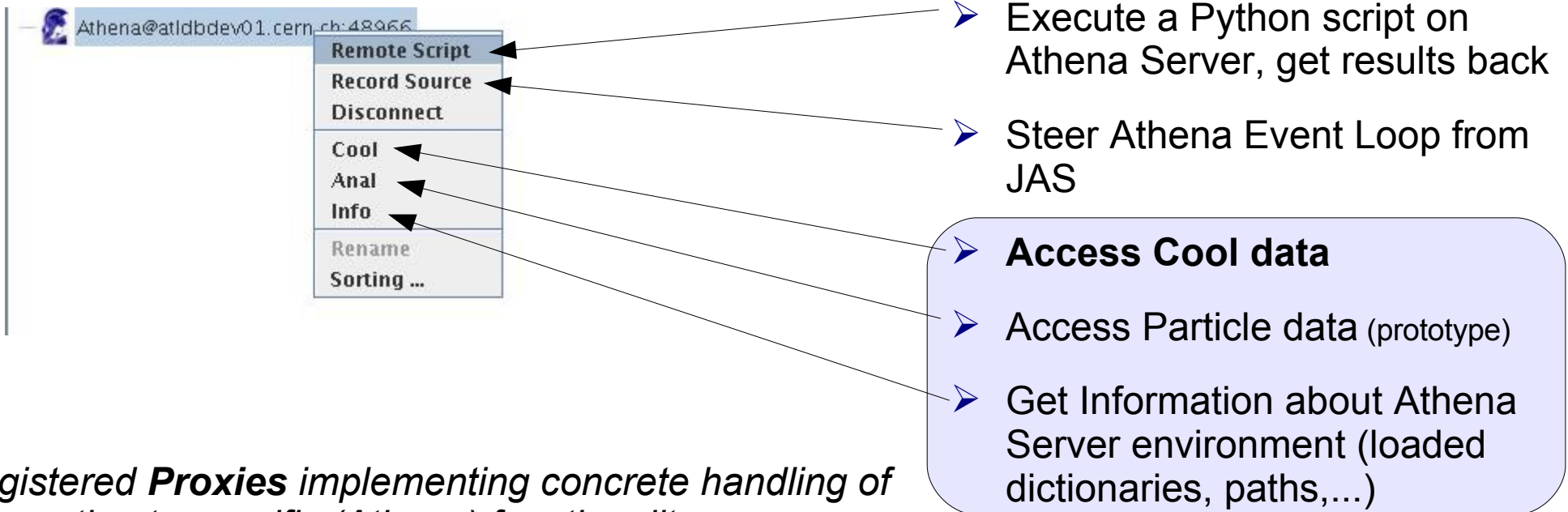
*On Server
(Linux with Athena)*

```
.....  
execfile ("InteractiveServer.py")  
server = InteractiveServer  
server.start()
```

*Server script written
by Atlantis team*



Interact with Athena



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

- 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.



Execute Python on Athena

Script to be executed on remote Athena

```
1 print self
```

Script Result

```
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



Steer Athena Event Loop

Next Event,...

The screenshot shows the JAS3 IDE interface. The top menu bar includes File, Edit, View, Tuple, Loop, Window, Athenaem, and Help. The toolbar contains navigation and editing icons. The left sidebar shows a file explorer with a folder named 'DataSets' containing a file 'Athena@lxplus003.cern.ch:48966'. The main editor window displays 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
```

The output console at the bottom shows the following text:

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

The status bar at the bottom indicates the classpath and memory usage: `classpath:/net/hep/atlas/Core/Athenaem/JAS3Plugin/doc-files/EventLoop.java` and `6.41/8.77MB`.

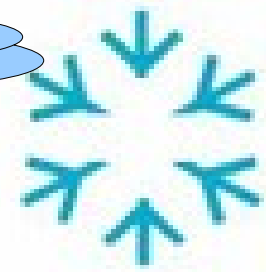
Athena interpreted as a set of Records

Python script executed on each Event

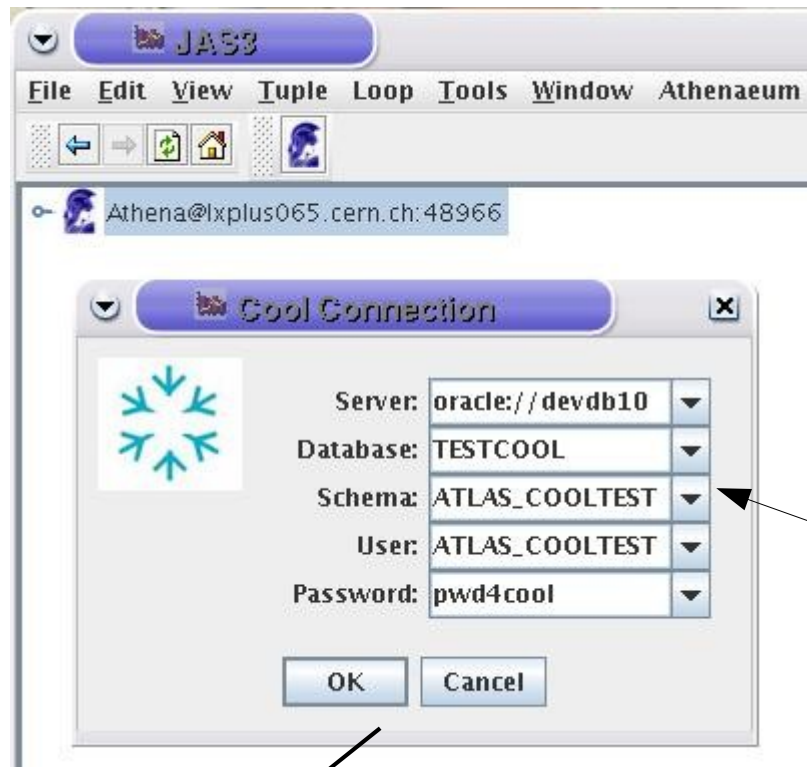
Output Console



LCG Conditions Database



Interact with Cool



JAS + Athena
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 Python script used to get data





Work with Cool

Athena@lxplus065.cern.ch:48966

Cool

tree-0

Cool@oracle:devdb10[TESTCOOL, ATLAS_COOLTEST]

IOVDbTest

- IOVDbTestAMDBCorrection
- IOVDbTestAttrList
 - since
 - until
 - object
 - channel
 - day
 - month
 - year
 - hourl
 - minute
 - second
 - xPosition
 - id
 - name
- IOVDbTestMDTEleMap
- IOVDbTestMDTEleMapColl

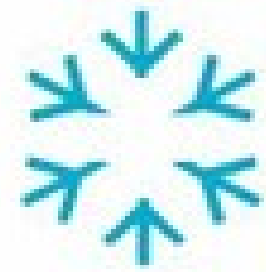
Indet

- Align
 - TRT_DF_B0
 - TRT_DF_B1
 - TRT_DF_B2
- Calib

Cool Proxy for Athenaem × Cool Result × IOVDbTestAttrList ×

| since | until | obj... | cha... | day | month | year | hourl | 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
 - (AIDA) NTuples
 - HTML
- and accessed
 - via GUI
 - using scripting interface (Java, Python, Pnuts)
 - using API (Java, Python)



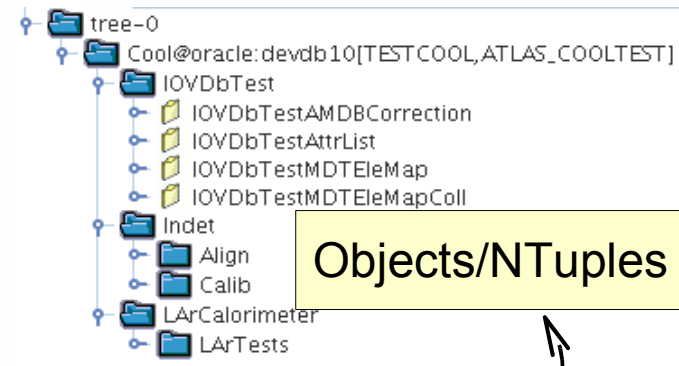
Cool XML

/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='[...]'/>
  </payload>
</folder>
  
```

XML View

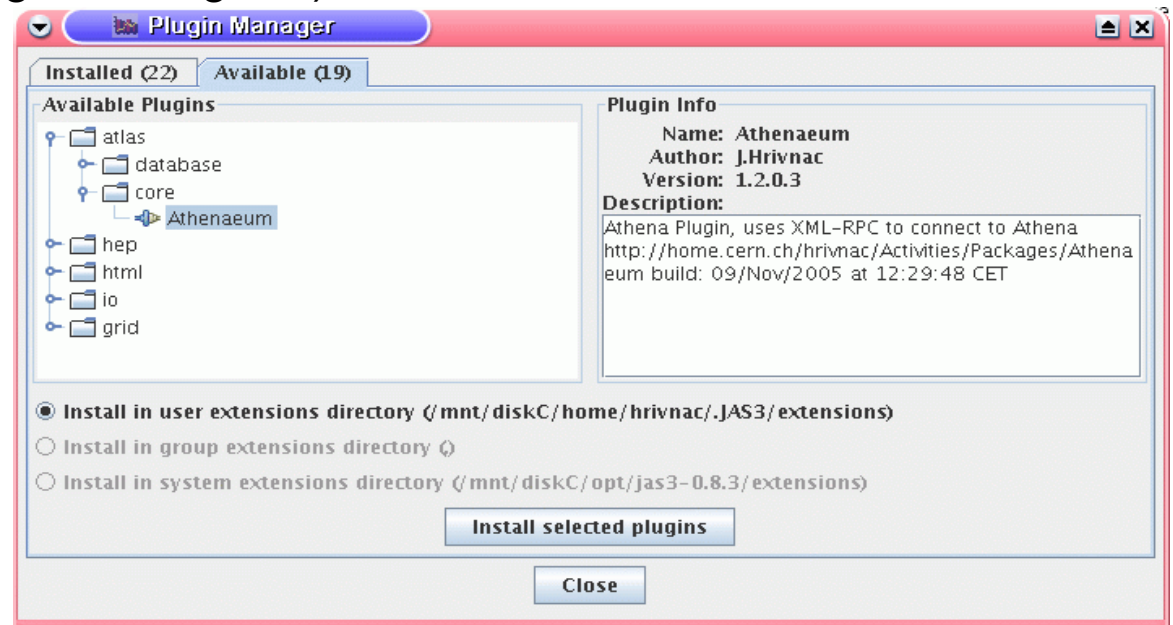
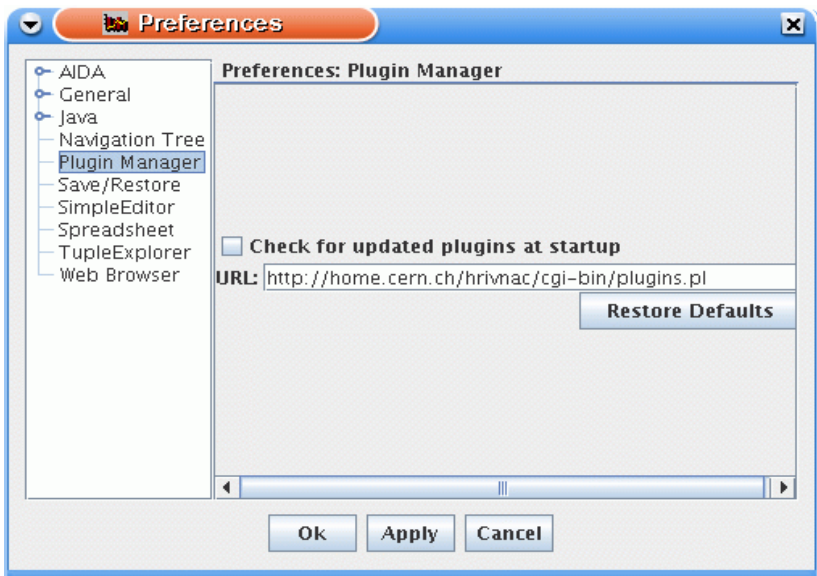
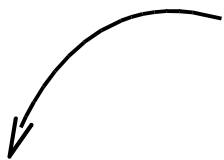
XML Schema

Python



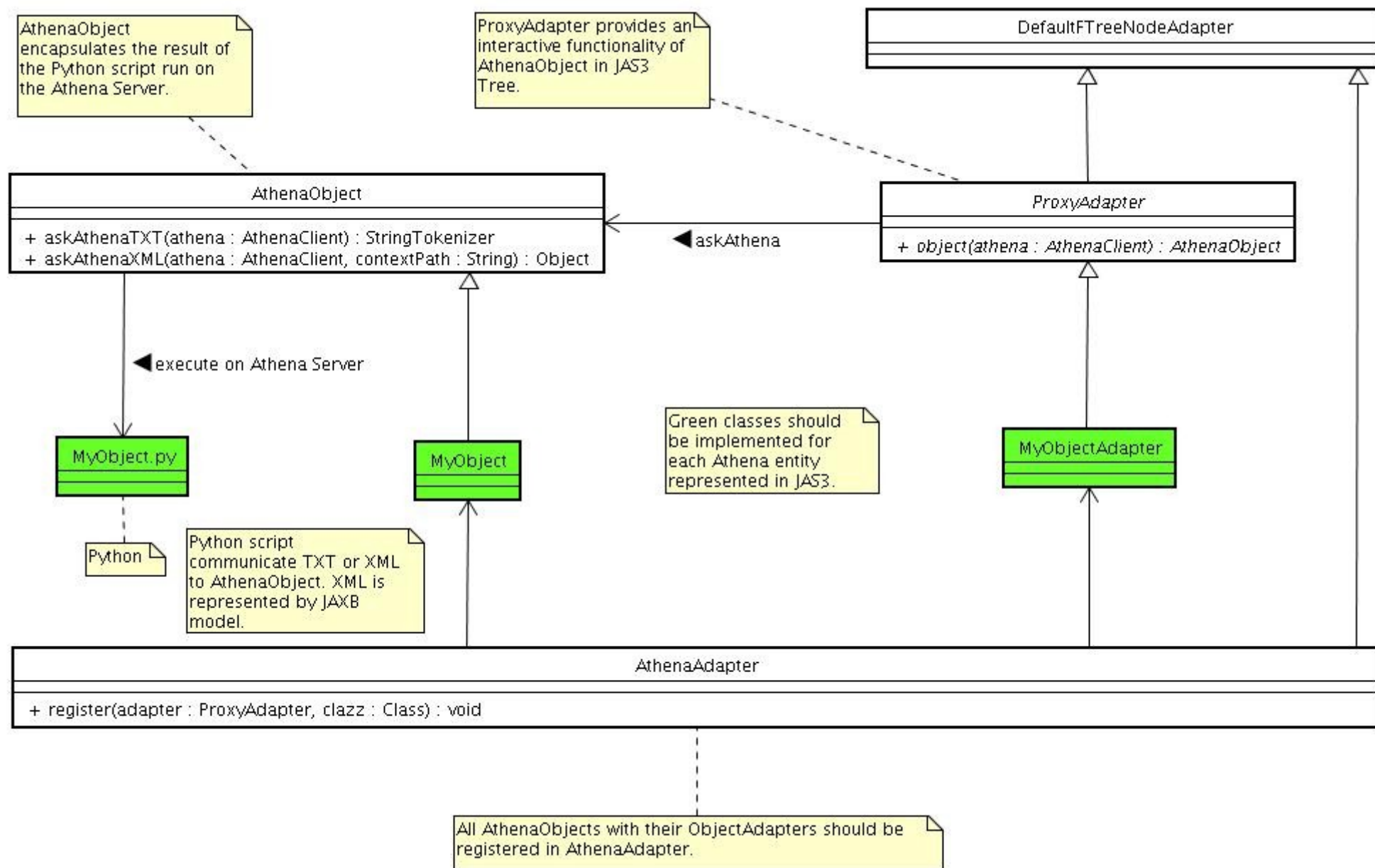
How To Start

- 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...)



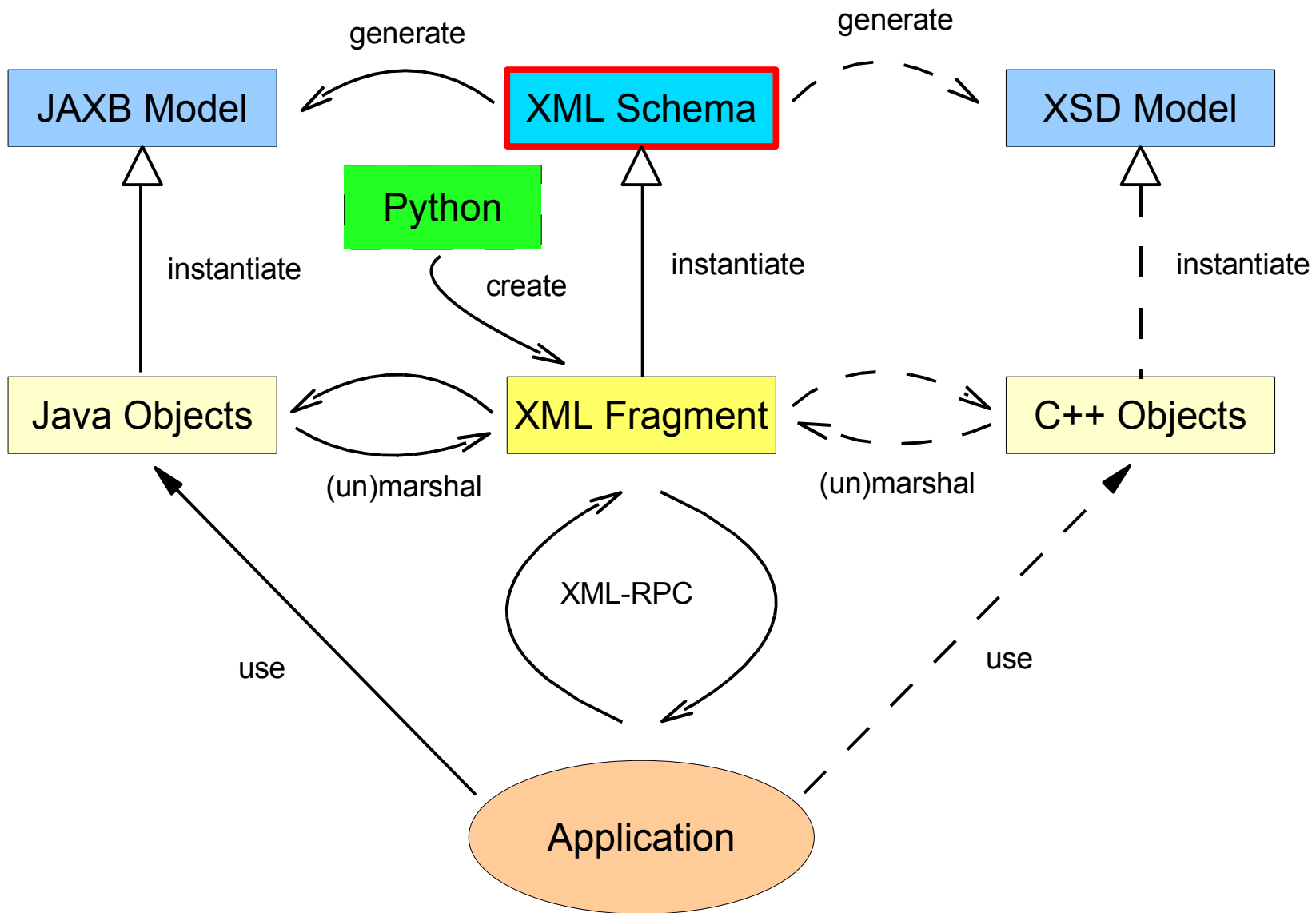


How To Write New Proxy



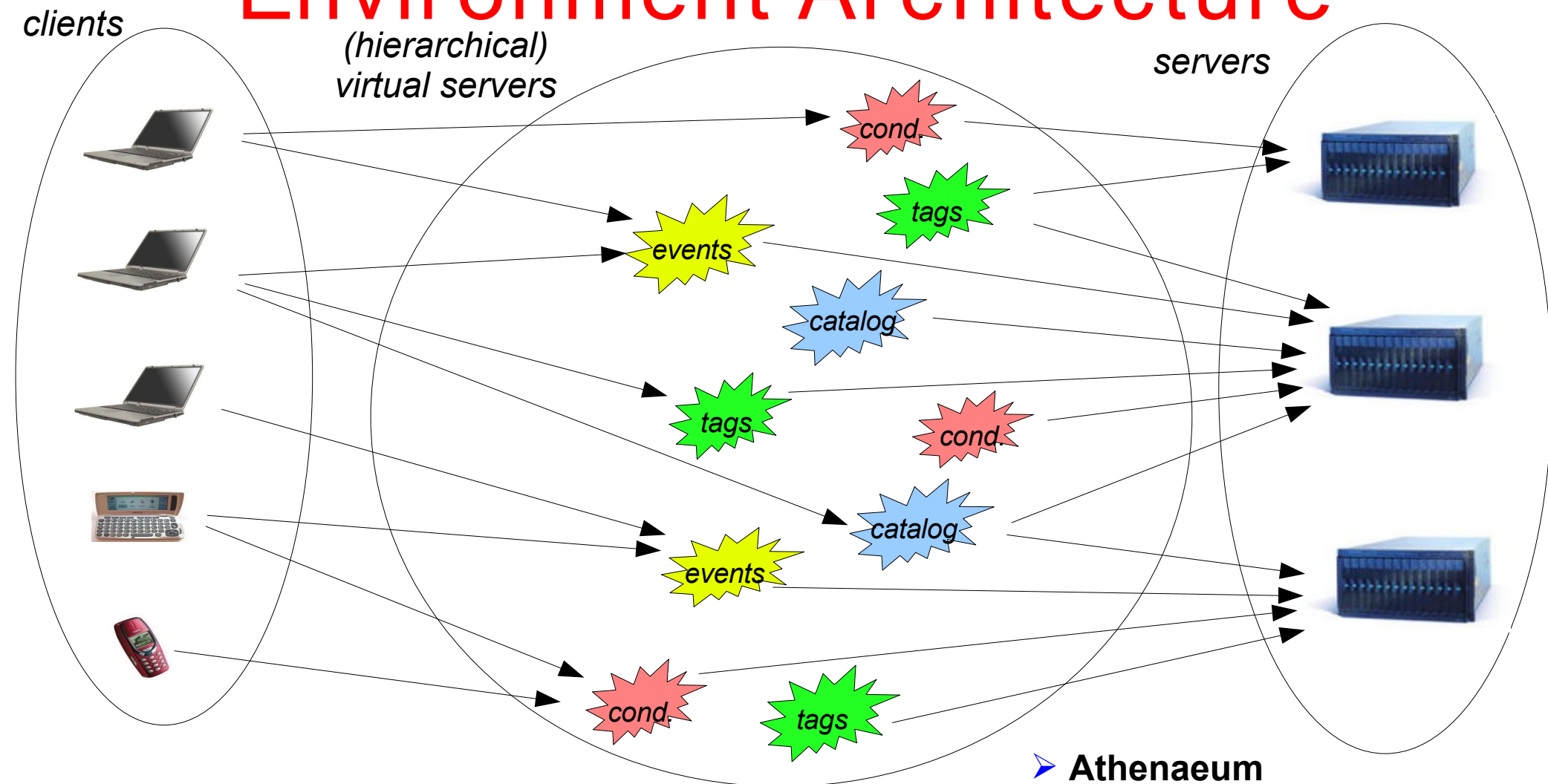


XML Schema Representations





Distributed Interactive Environment Architecture



- 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
- EJB
- (CMS) Clarens



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)
- 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

➤ Athenaeum

- Lazy & Compressed data transport (to speed up)
- Generic XML GUI
- User-customizable XSLT
- More Proxies (Analysis objects, Generic StoreGate access, ...)
- Athena startable from Athenaeum (remotely)
- Deployment of *Athena Servers*
- Merge of the Server script changes back to Atlantis script

➤ Cool Browser

- Interpretation of Tokens & AttributeLists
- Query with Tags & Time Intervals
- Merge XSD & XSLT with Shaun Roe
- Better Graphics
- Possibility to write/update Cool DB



Help



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



Athenaeum - Athena JAS3 Plugin

- [How to Use JAS](#)
- [How To Start Athena Python Server](#)
- [How To Connect to Athena Python Server](#)
- [How to Work with Local Scripts](#)
- [How to Work with Remote Scripts](#)
- [How to Loop over Events](#)
- [How to Work with Proxies of Athena Objects](#)
- [How to Write Proxies of Athena Objects](#) (for experts)
- Existing Proxies:
 - [Info](#) (example proxy)
 - [Analysis](#)
 - [Cool](#)
- [How to Use in a Standalone Environment](#)
- [Where to Find More](#)

How to Use JAS

The JAS3 documentation is [available](#). JAS3 also contains its integrated help system with executable examples.

How To Start Athena Python Server

Athena XML-RPC Server can be started using simple [Python Script](#) (this script has been developed by the [Atlas](#) team):

- Place the script into you run directory.
- Put following lines at the end of your joboptions.py script:

```
# initialise Athena and go the first Event
# (following two commands can be omitted here
# and called remotely once the Server is started)
theApp.initialize()
theApp.nextEvent()
# load the Server
execfile ("InteractiveServer.py")
# instantiate the Server
server = InteractiveServer()
# start the Server
server.start()
```
- Call Athena using following command:

```
athena.py -i jobOptions.py -s
```
- The message with the Server URL will be printed out. It should be used for the connection from JAS.

How To Connect to Athena Python Server

5.26/8.56MB