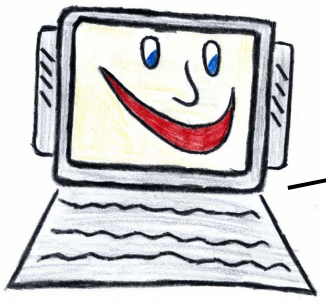
Athenaeum

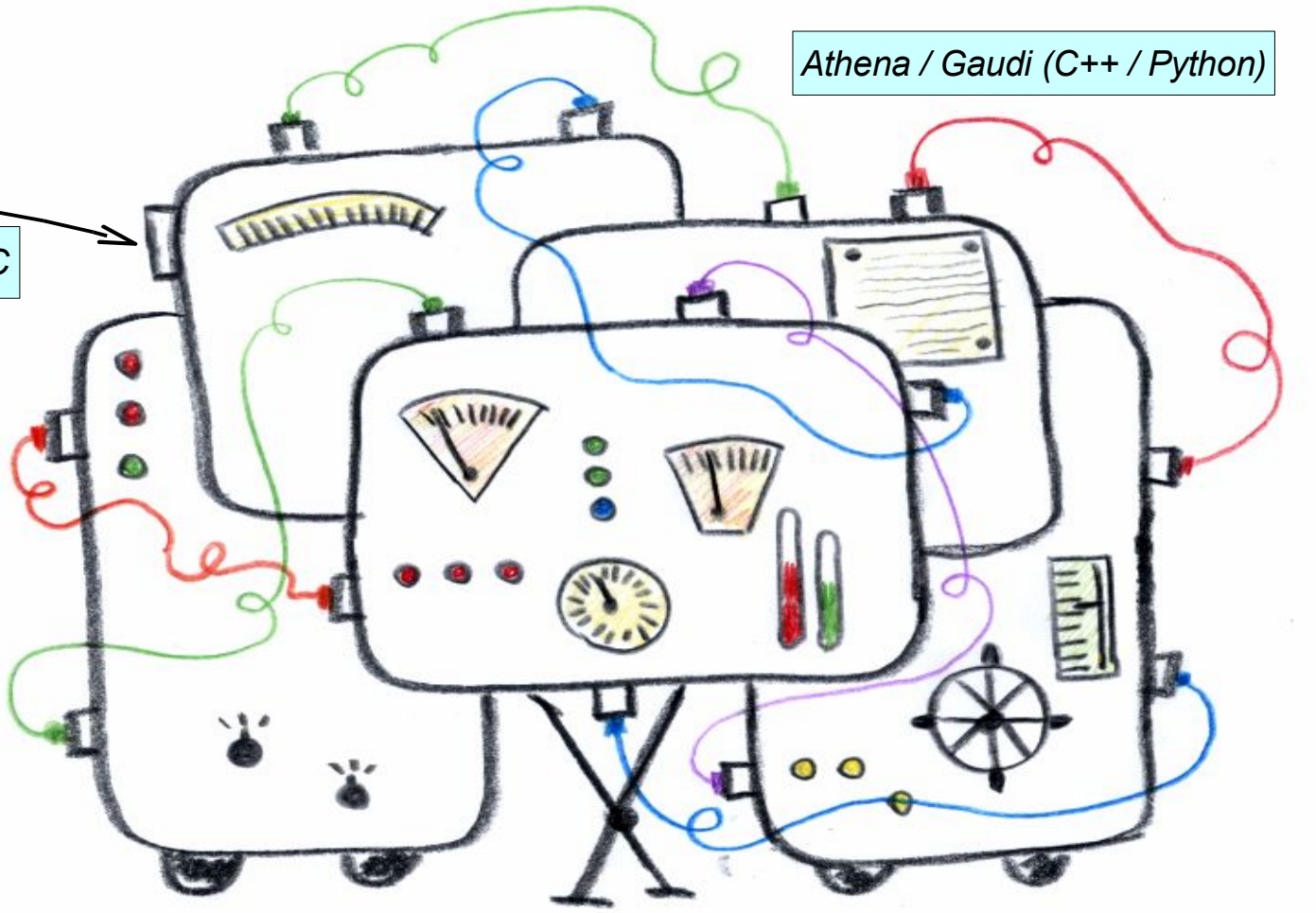
Using Java Analysis Studio as an interface to the Atlas Offline Framework



Athenaeum / JAS (Java)

XML-RPC

Athena / Gaudi (C++ / Python)



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

J.Hrivnac, LAL/Orsay
CHEP'06/Mumbai, Feb'06

Athenaeum

Using Java Analysis Studio as an interface to the Atlas Offline Framework

Local Client

Athenaeum Plugin

JAS
Java Analysis Studio

FreeHEP (SLAC)

Java / xMB
(runs everywhere)

XML

XML-RPC

Python

(Remote) Server

XML-RPC Server

Python API

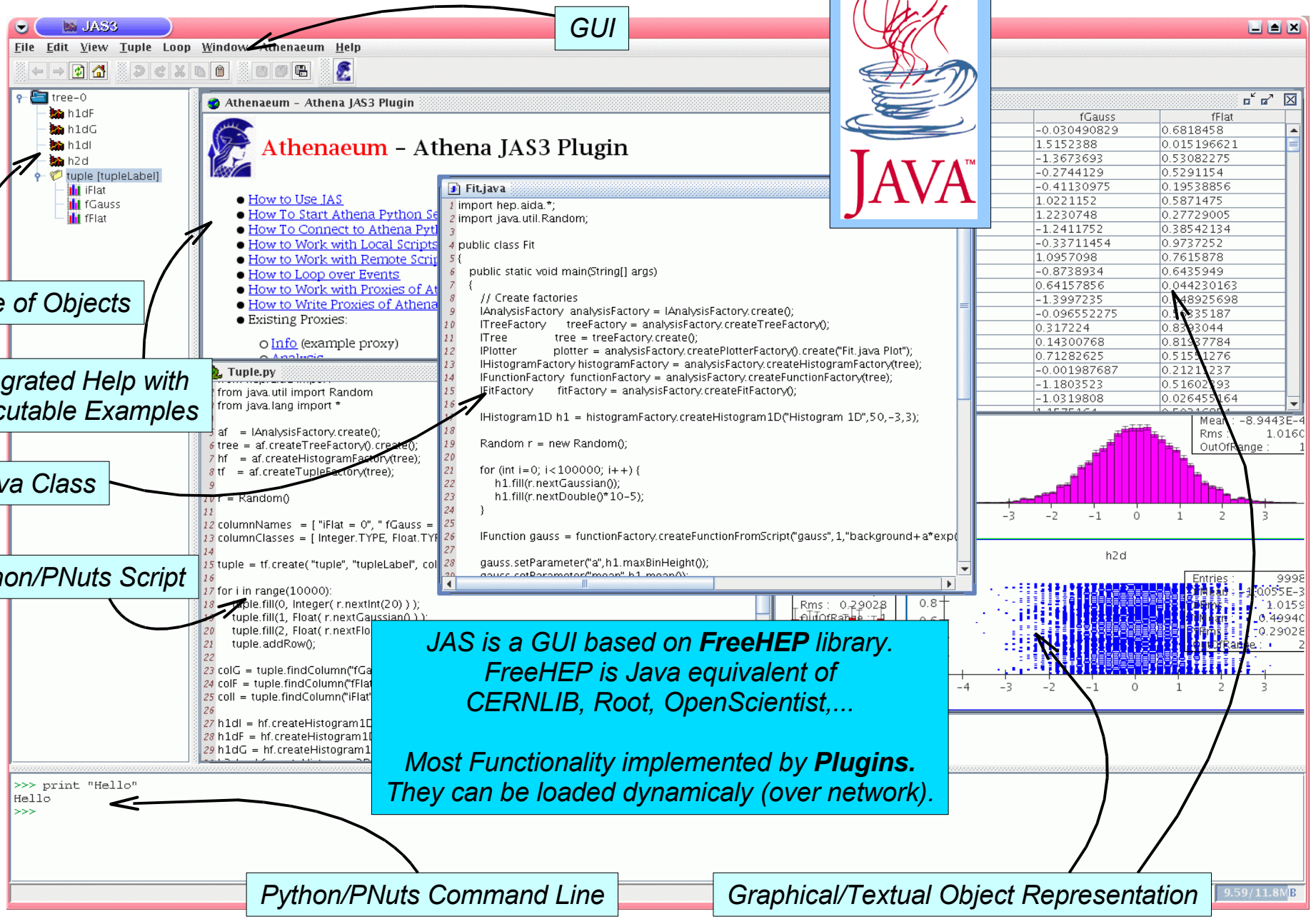
Athena / Gaudi
Atlas Offline Framework

LCG (CERN)

C++ / xGB
(runs only on lxplus)

- ***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 within JAS.*

Java Analysis Studio



Atlas Offline Framework (C++ / Python)

Open Connection to Athena

On Client
(Any platform with JAS + Athenaeum 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”)

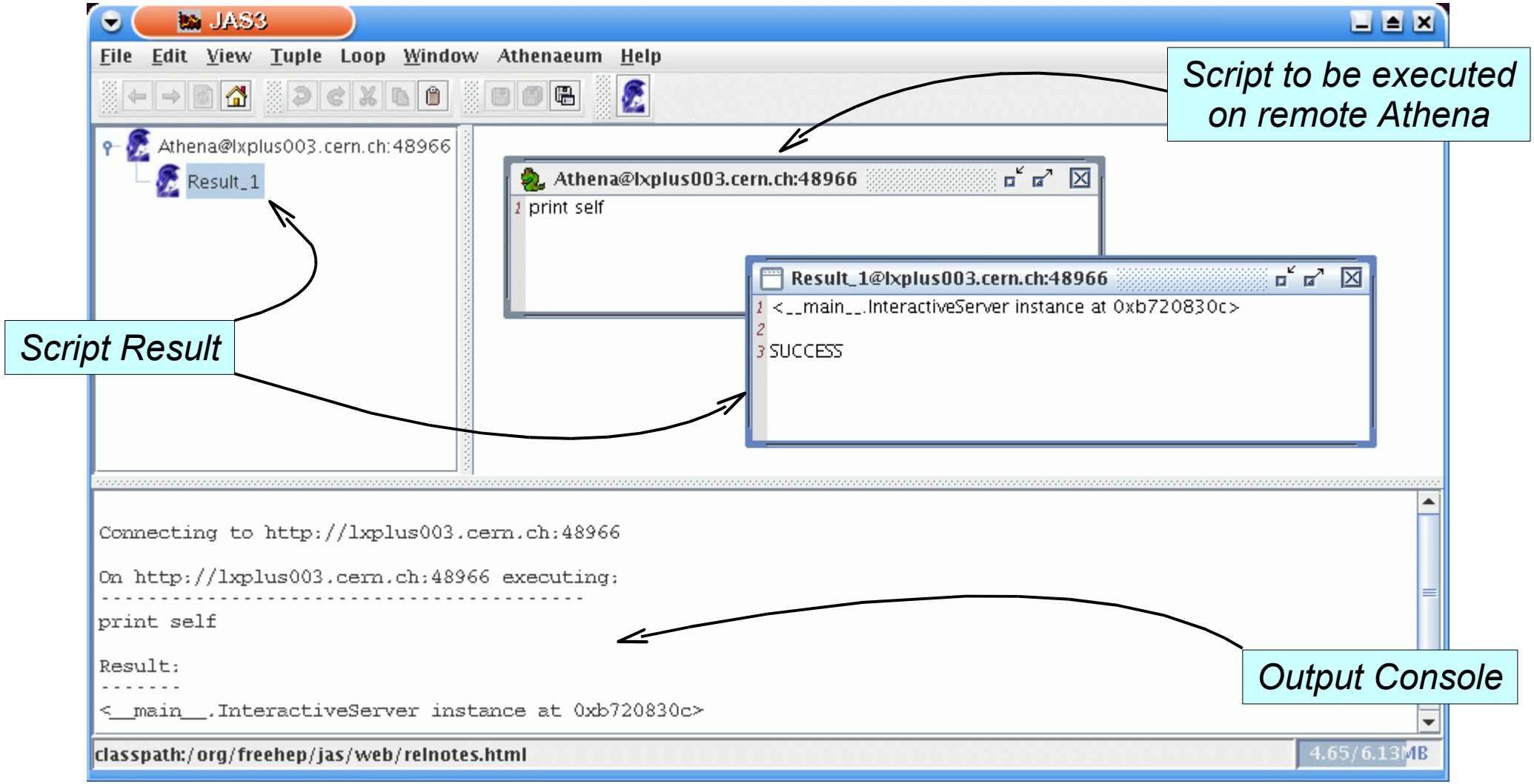
Server script written
by Atlantis team
(http://cern.ch/atlantis)

Interact with Athena

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

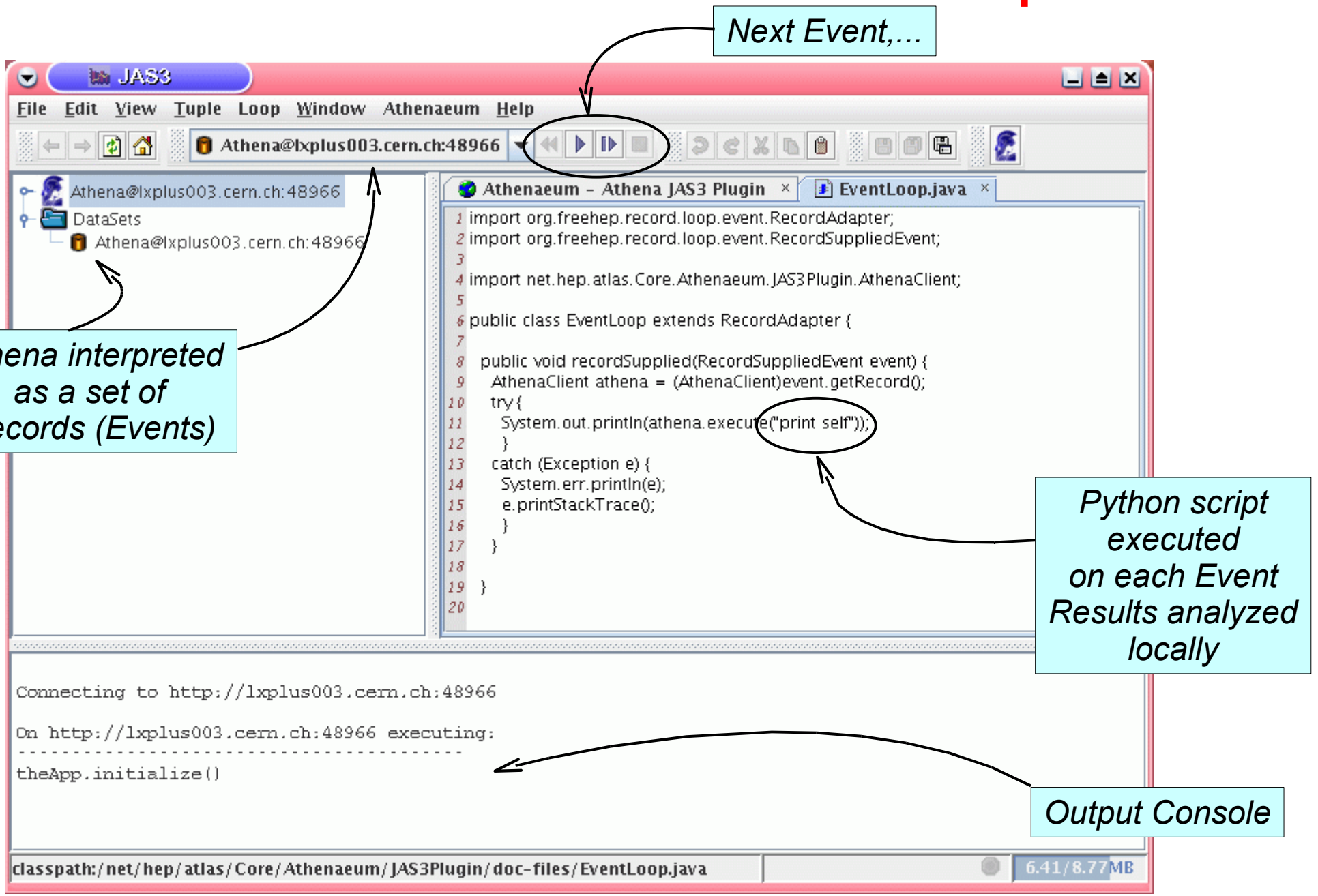
*Registered **Proxies** are implementing concrete handling of connection to specific (Athena) functionality.*

Execute Python on Athena



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



Remote Proxy

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

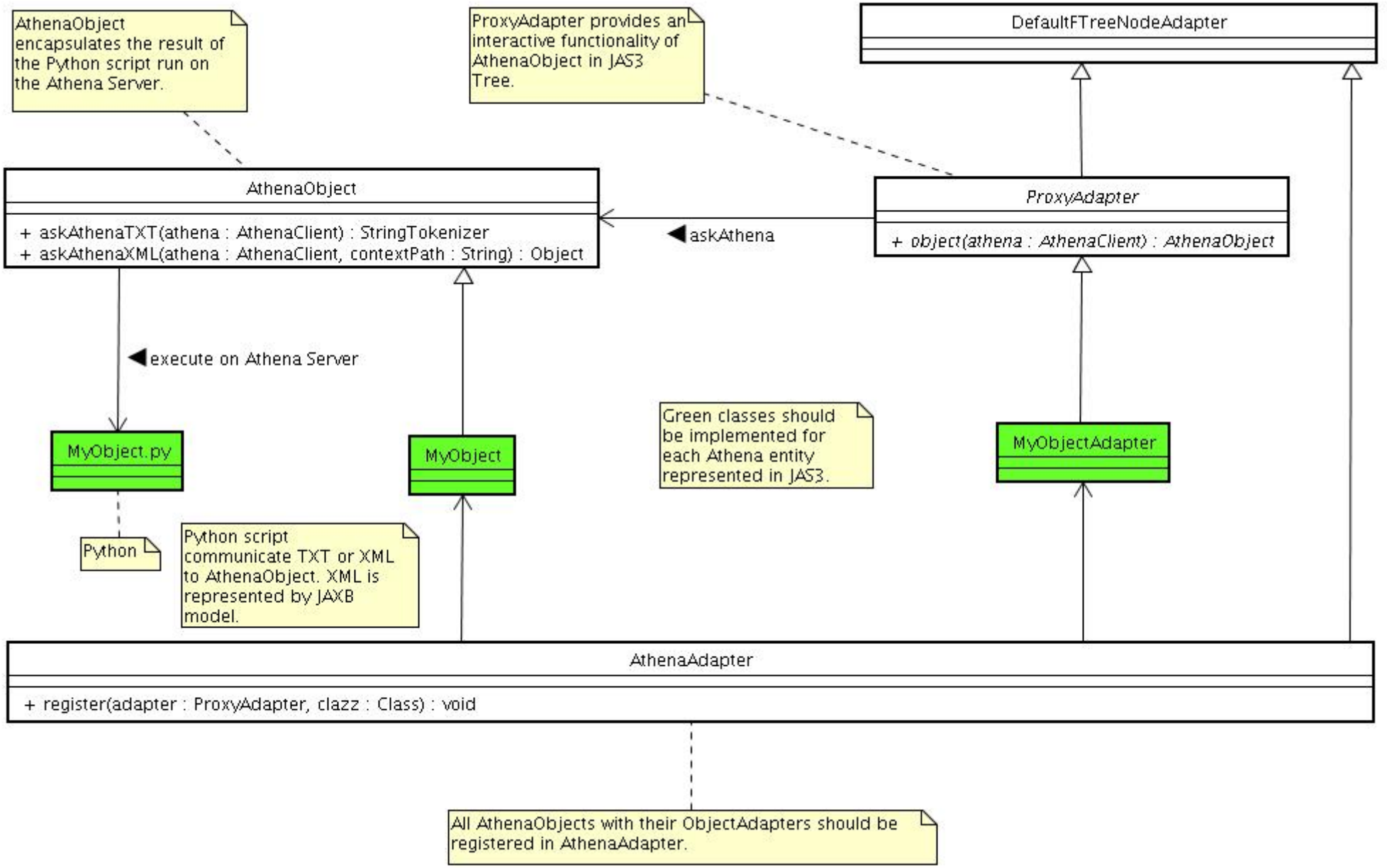
- Athena Python script to extract data from Athena

- JAS wrapper to present/handle data inside JAS

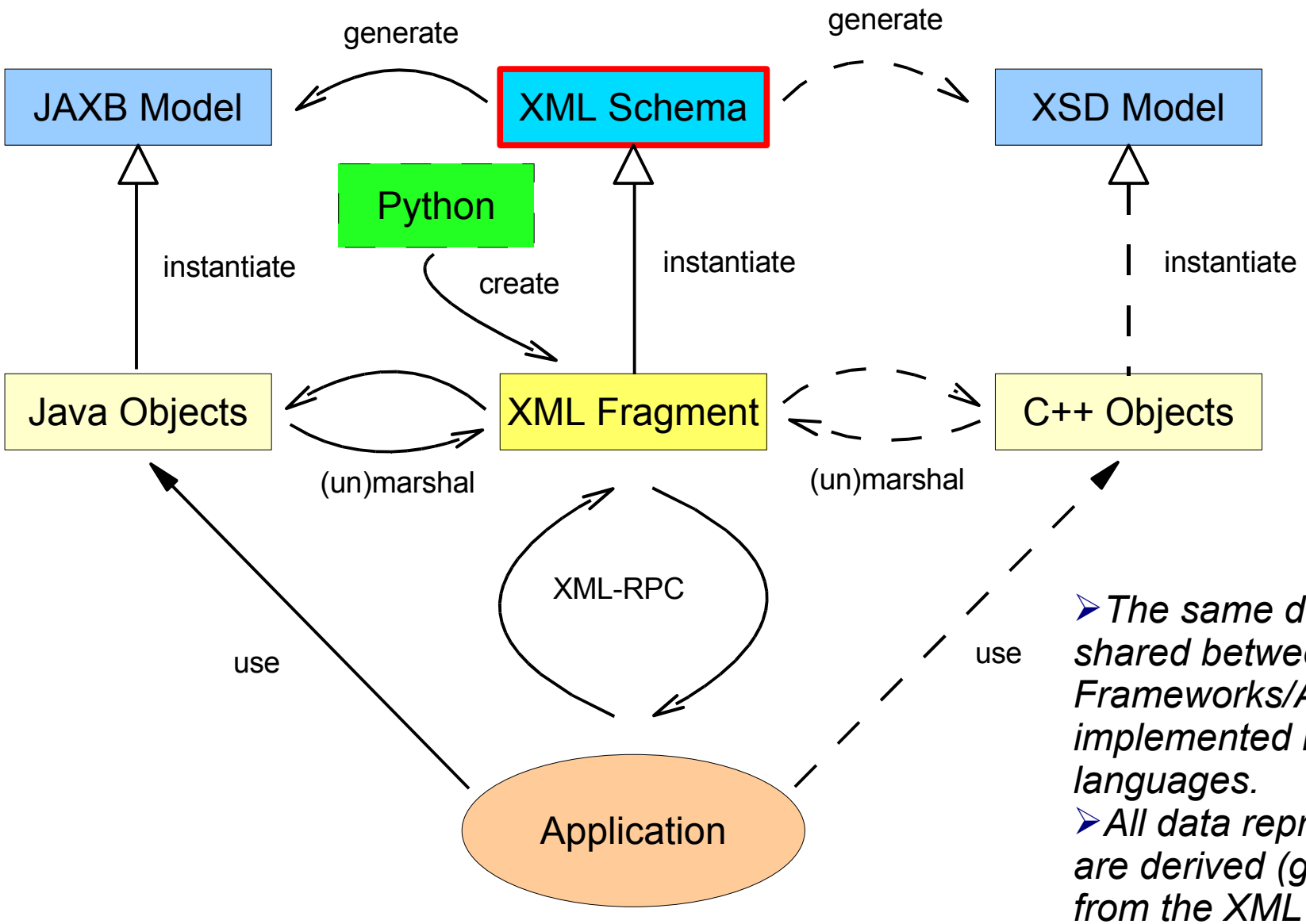
- XML schema to describe data

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.*

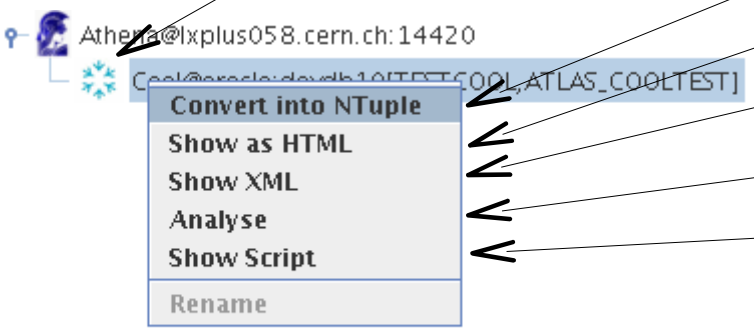
Interact with Cool

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

JAS + Athenaeum Client

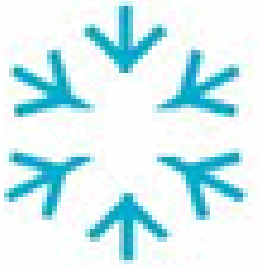
Athena/PyCool Server

Cool DB Server



- Open connection to Cool DB
- Interpret data (as AIDA NTuples)
- Show data as HTML
- Show data as XML
- Analyse data
- Show Python script used to get data

Cool XML

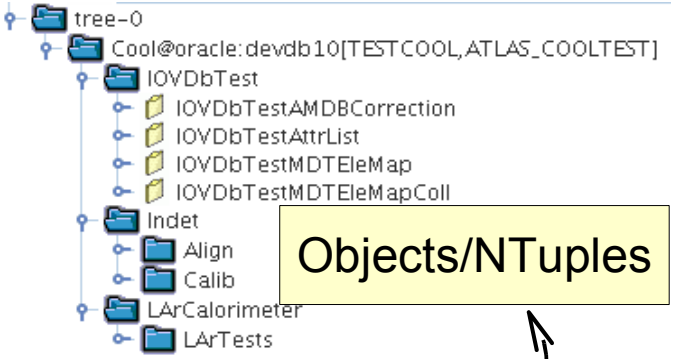


/IOVDbTest/IOVDbTestAttrList[1326]

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

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

HTML View



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>
```

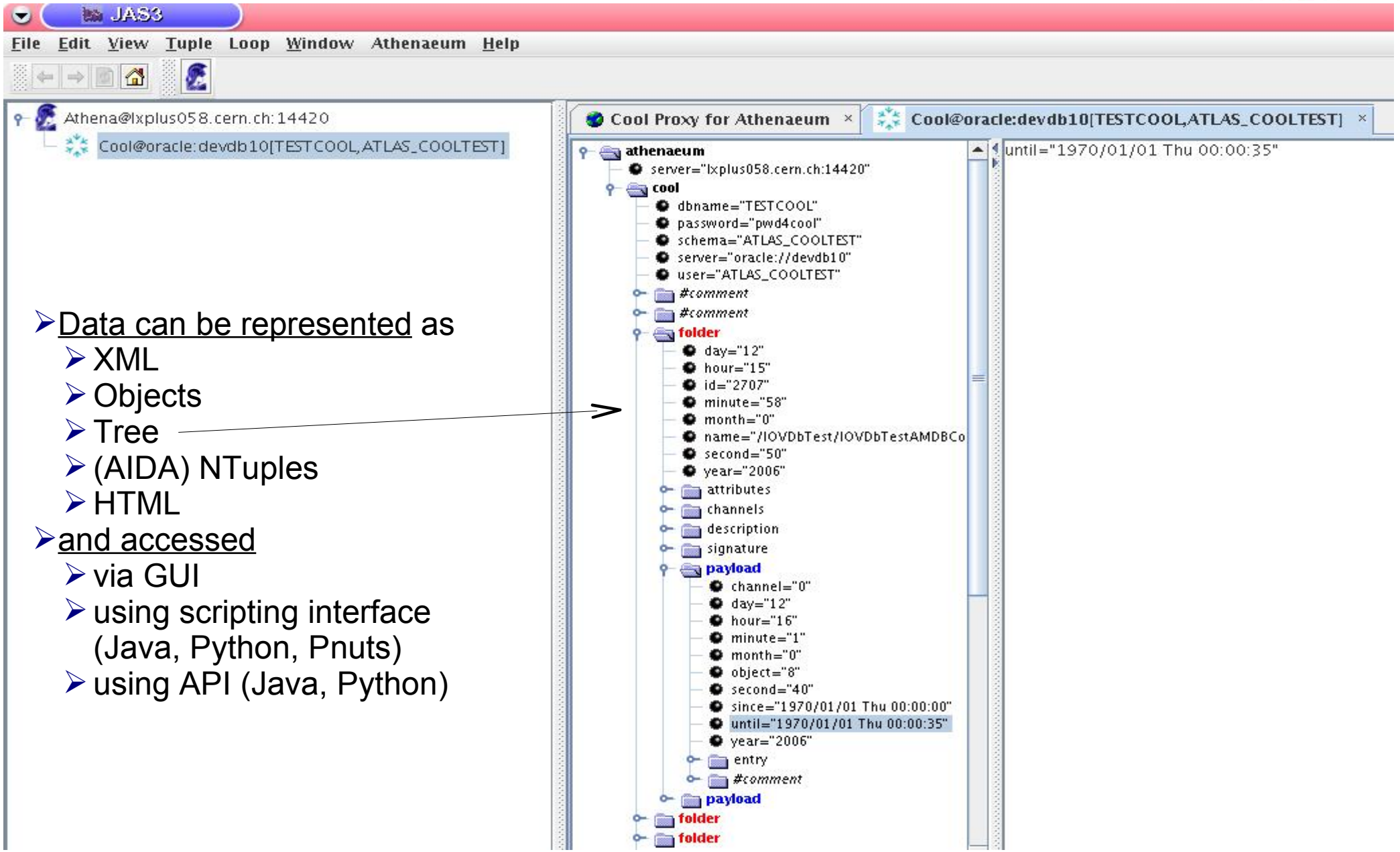
XML View

XML Schema

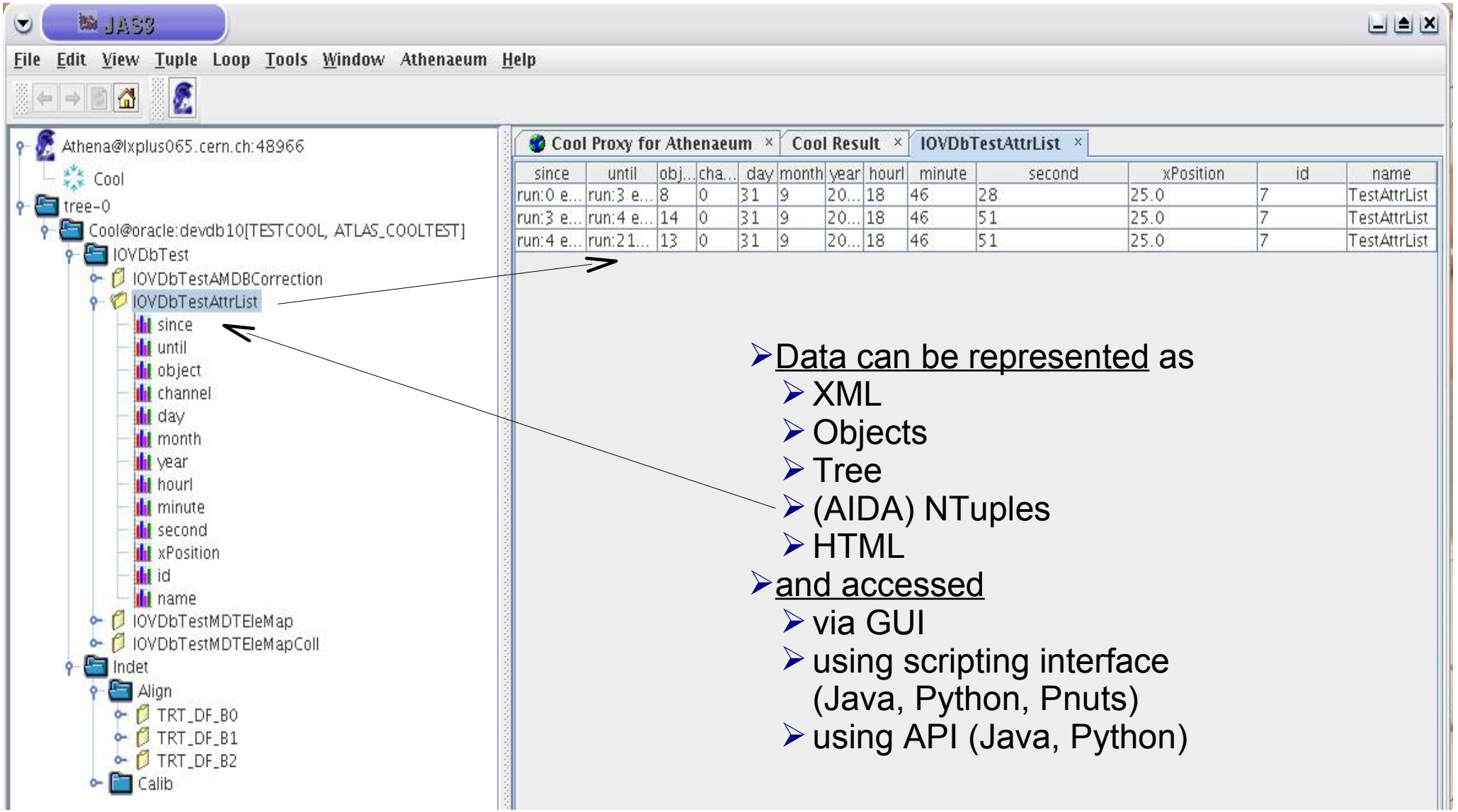
Python

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)

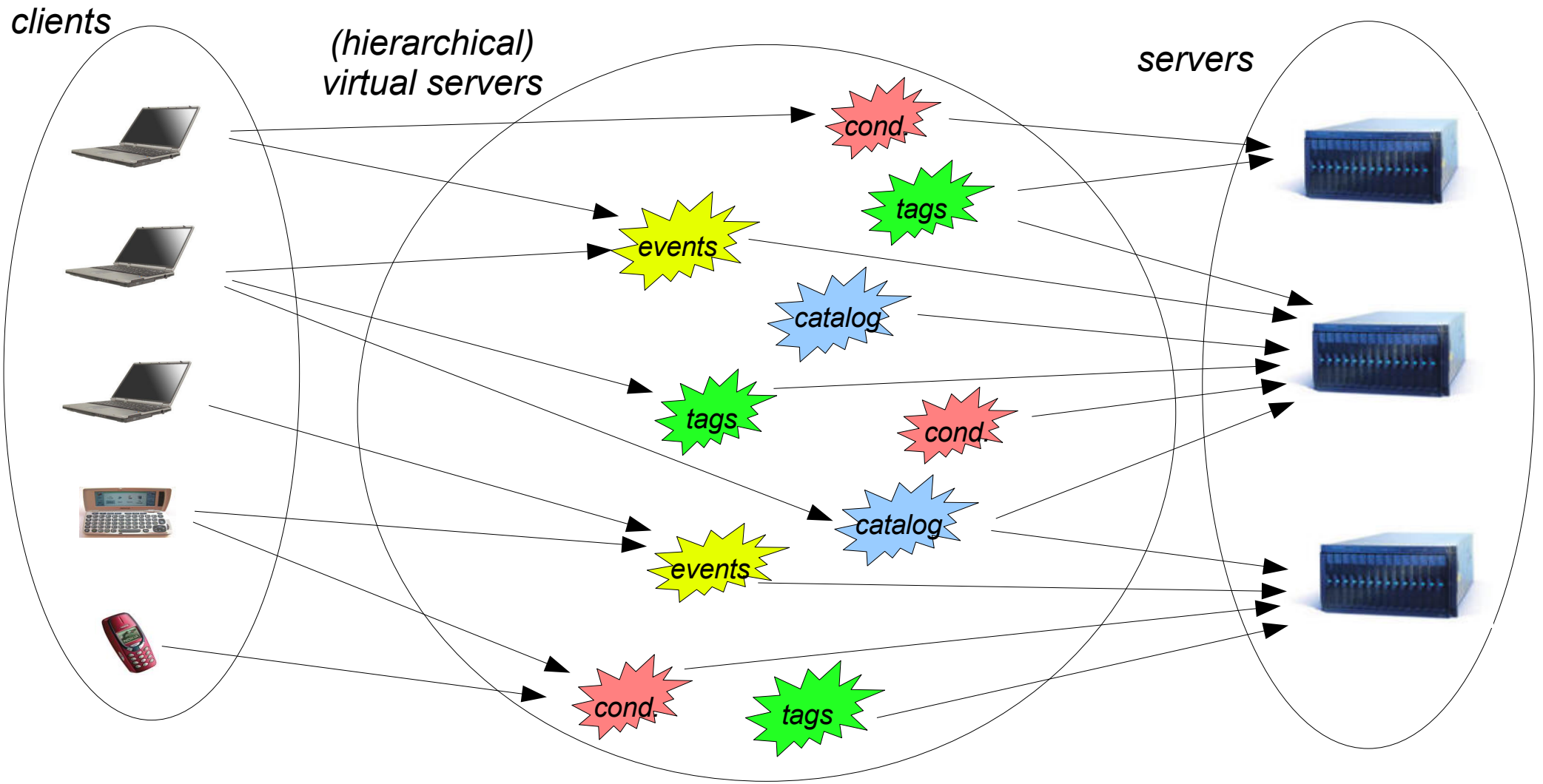


Work with Cool (2)



- 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)

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 (see poster 331)
- Sequoia (see poster 331)

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

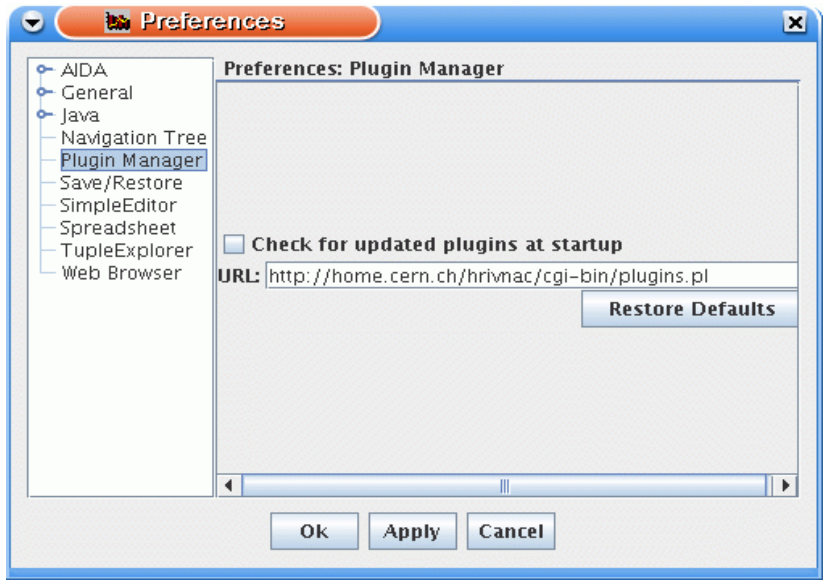
- PyAthena (Python API to Athena)
  - **Incomplete** (only a subset of C++ API is available via Python)
  - **Undocumented** (C++ Doxygen is not enough for documentation of its Python API; it is not easy to guess the meaning of weakly-typed methods; code fragments on Web/Wiky are often out-of-date)
  - **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

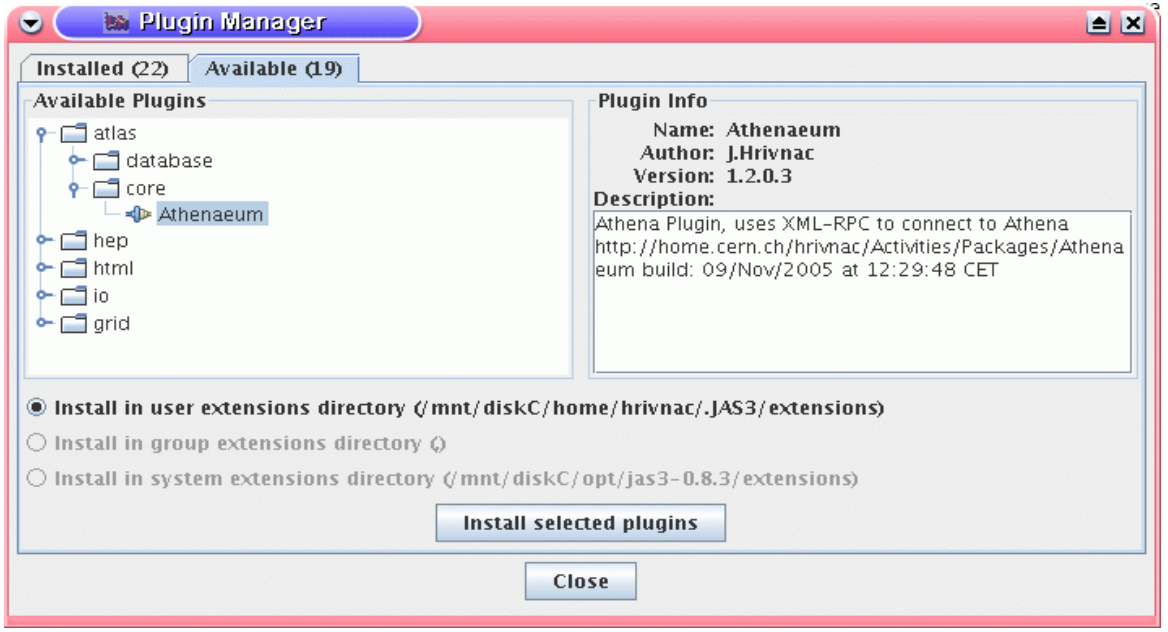
To Do Next

- 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
- User-customizable XSLT
- 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...)





Help

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