

GeodiseLab: Making the Grid Usable

Graeme Pound
UK e-Science All Hands Meeting
19 - 22nd September 2005

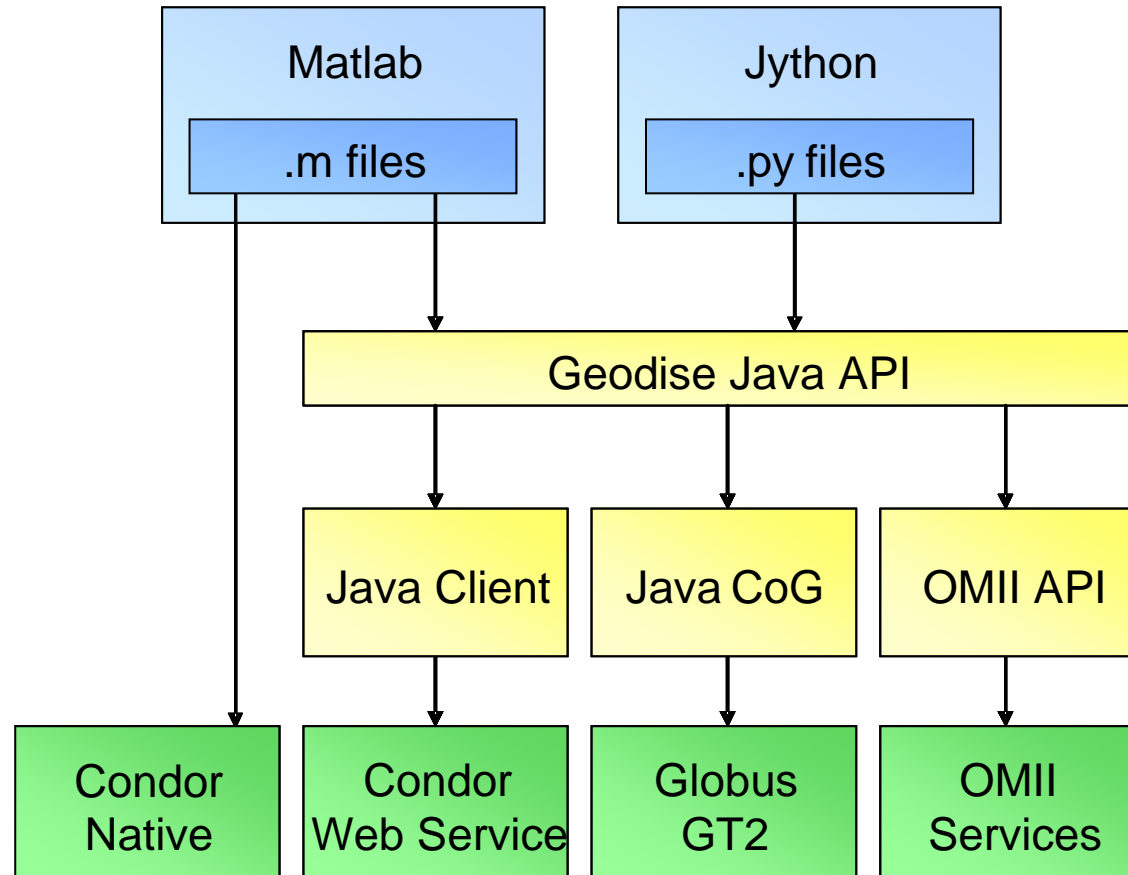


GeodiseLab

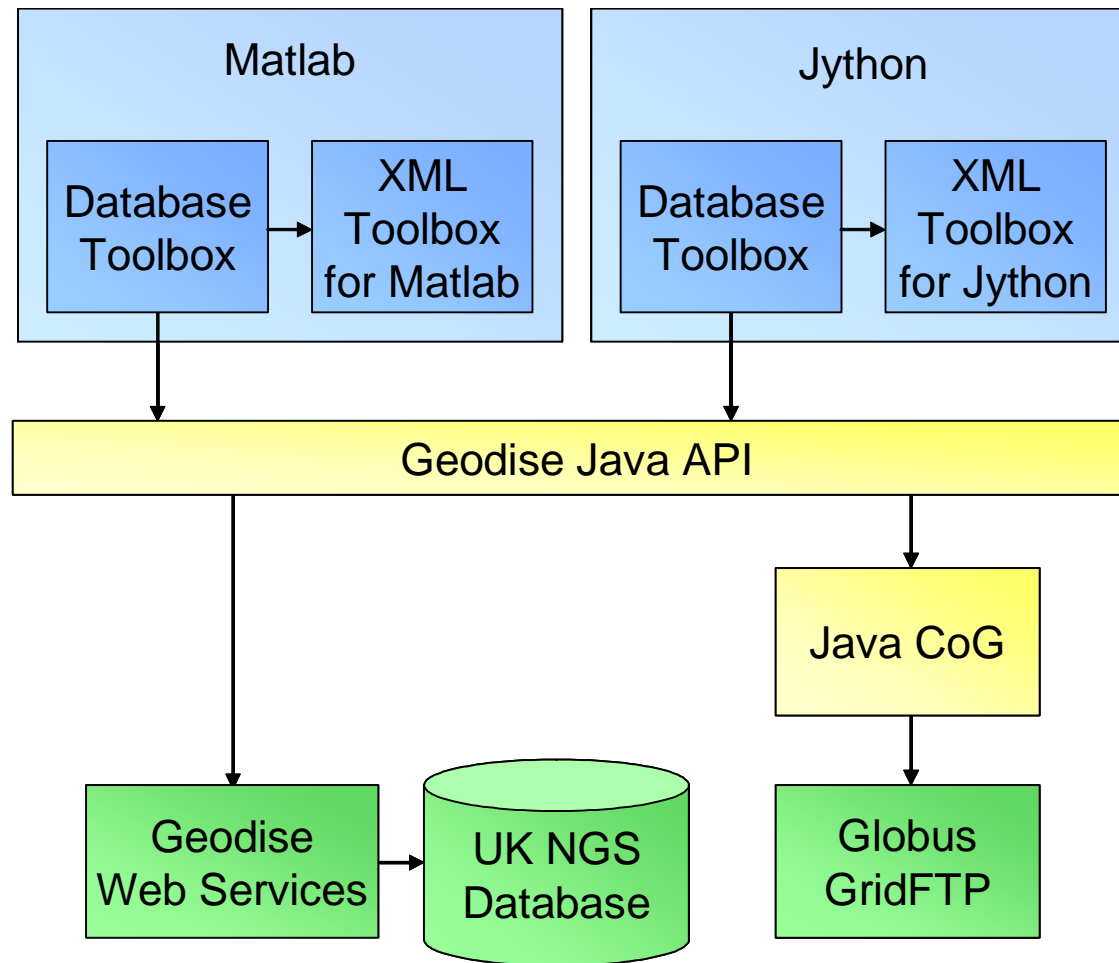
- Bringing the Grid to the engineer
- Matlab - technical computing environment
 - 500,000+ users
 - Data analysis and visualisation toolboxes
 - High-level scripting
- Jython - Java implementation of Python
 - object-oriented Python language
 - 100% pure Java
 - Active funded development
- Scripting complex engineering workflows
- Tutorials, Workshops and Support



Computational Toolboxes



Database and XML Toolboxes



Development

- Response to user feedback
 - bug fixes
 - feature requests
- Robustness testing
- Feature enhancements
 - new data types (XML Toolbox)
 - mark for deletion (Database Toolbox)
 - query constraints (Database Toolbox)
- Documentation and tutorials
 - valuable feedback from QA
- Jython support
 - support for Python syntax
- New Computational Toolboxes
 - OMII Toolbox
 - CondorNative Toolbox



Data Transfer between PSEs

Matlab

```
>> a.b = 3.1415926535897;
>> a.c = 'a character string'

a =

    b: 3.1416
    c: 'a character string'

>> xml_save( 'demo.xml', a )
>> type demo.xml

<?xml version="1.0"?>
<!-- Written on 01-Jul-2005 13:54:45 using the XML Toolbox for Matlab
-->
<root xml_tb_version="3.1" idx="1" type="struct" size="1 1">
  <b idx="1" type="double" size="1 1">3.1415926535897</b>
  <c idx="1" type="char" size="1 18">a character string</c>
</root>
```

Jython

```
>>> from gdxml import xml_load
>>> v = xml_load( 'demo.xml' )
>>> print v
{'b': 3.1415926535897, 'c': 'a character string'}
```



GEOS

GeodiseLab Releases

- Compute, Database and XML Toolboxes
 - Globus GT2 client functionality
 - managed data archive
 - transparent XML handling
- GeodiseLab for Matlab 1.2.0
- GeodiseLabPy for Jython 1.0.1



GeodiseLab Users

- **Computational Engineering and Design Centre**
 - variety of engineering applications, including aircraft and F1 design, structural analyses
 - feedback from user workshops
- **GENIE project**
 - Grid Enabled Integrated Earth system model
 - Data archive leveraged for structured data
- **XML Toolbox**
 - 1566 downloads from MatlabCentral in 18 months
 - User base in academic & commercial research



GeodiseLab Server

- Geodise Database Server 0.2.0
- Web services deployed on:
 - Tomcat 4.1
 - WebSphere Application Server 5.x
- Database
 - Oracle 10g or 9i
- Documentation
 - Deploying GeodiseLab web services and database schema
 - WebSphere and Tomcat installation
 - Oracle 9i and 10g installation
 - Globus installation
 - Obtaining a server certificate
- Awaiting internal SeSC QA



OMII Toolbox

- OMII Toolboxes for Matlab and Jython
 - client functionality to OMII 1.2
- Simplifies interaction with the OMII platform
- Functions for...
 - the management of OMII accounts
 - the management of OMII resource allocations
 - the management of OMII data
 - the management of OMII jobs
 - keystore management
 - miscellaneous OMII functions
- Tutorials and function documentation
- Awaiting internal SeSC QA



CondorNative Toolbox

- High level Matlab functions for Condor job submission
- Abstracts interaction with Condor executables
 - Condor version 6.6.7 or later
- Used for analysing EIT and EEG data at UCL
- Awaiting internal SeSC QA



EIT example (part I)

```
% create a number of 3D head models, distorted by a small factor
distortion_factor = {0.0,0.01,0.02,0.03,0.04,0.06,0.08,0.10,0.12};
for i=1:9
    model = cfg_load( 'default_head_3D.cfg' );
    model = stretch( model, distortion_factor{i} );
    inputfilename = generate_inputfile( model );
    metadata.distortion_factor = distortion_factor{i};
    metadata.model_name = 'head';
    gd_archive( inputfilename, metadata );
end

...
```



EIT example (part II)

```
% find all previous runs in database which match range
models = gd_query( [' model_name=head & ' ...
                   ' distortion_factor>=0 & ' ...
                   ' distortion_factor<0.20 ' ] );

for c = 1:length(models)
    % download file to local model directory
    filename = ['./model_', num2str(c), '.geom'];
    gd_retrieve( models{c}.standard.ID, filename );
    % create submission information
    job = condor_job( 'beginner' );
    job.executable = 'reconstruct3d.exe';
    job.arguments = filename;
    jobhandles{c} = condor_submit( job );
end

% wait for results
condor_waitfor( jobhandles );

% extract reconstruction errors
ErrList = extract_image_errors3d( './model_*', ...
                                  'default_head_3D.cfg' );
```

