Geodise:
Taking the Grid to the Engineer

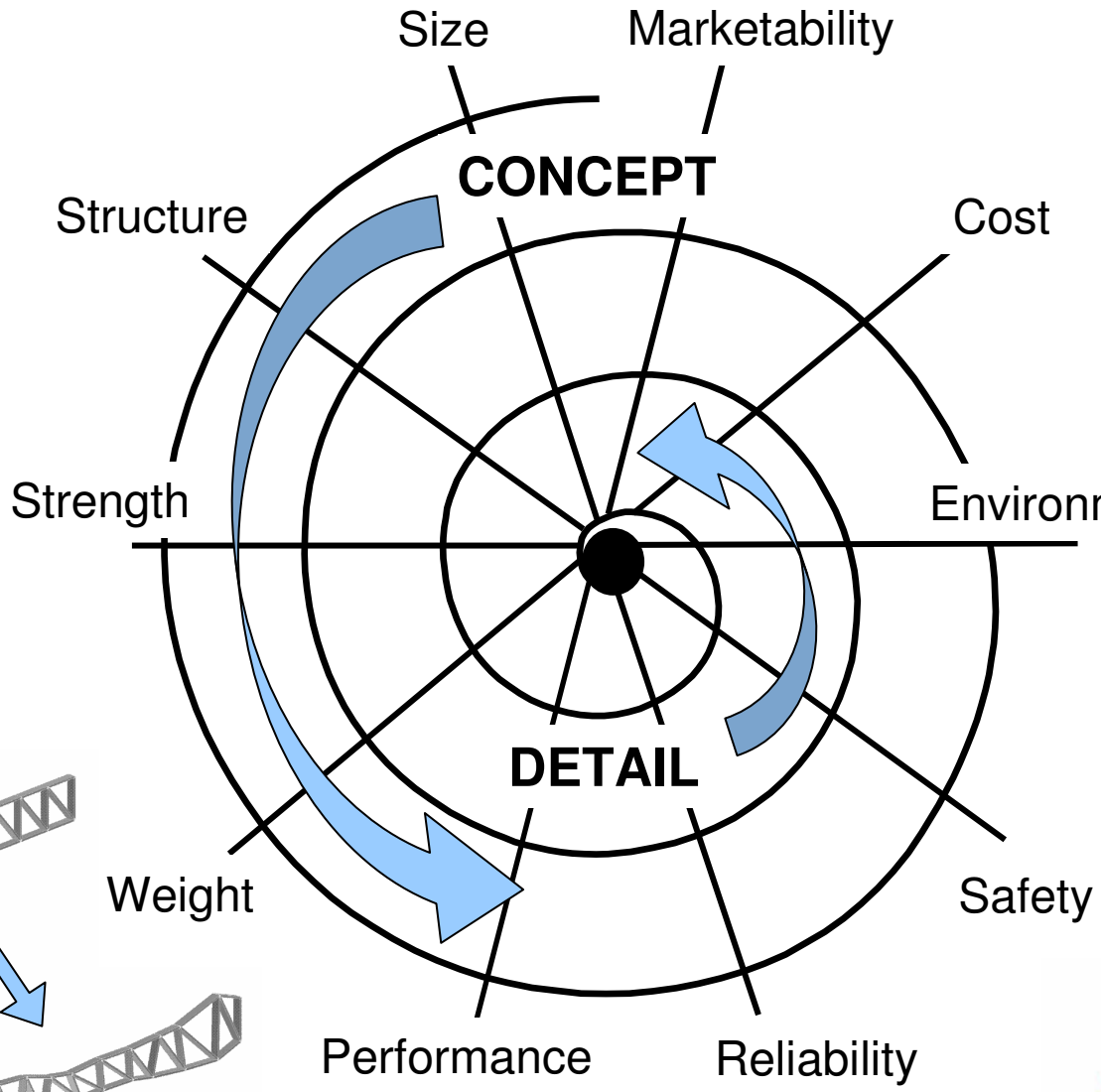
Graeme Pound

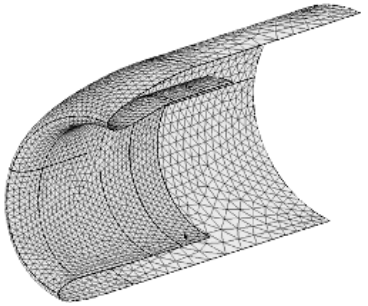
International Summer School on Grid Computing 2006
Ischia, Italy 9th-21st July 2006

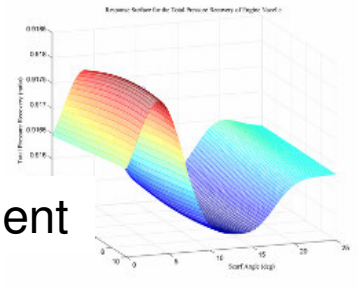
Lecture summary

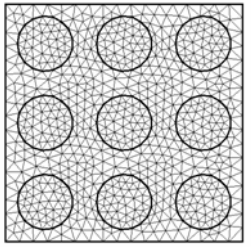
- Application domain
  - Engineering design search and optimisation
  - Computational Fluid Dynamics
- Design principles
  - Usability
  - Integration
  - Share and reuse
- Technical solution
  - Generic toolboxes
  - Job submission
  - Data management
- Application Examples
- Conclusions

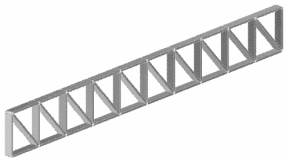
Application Domain

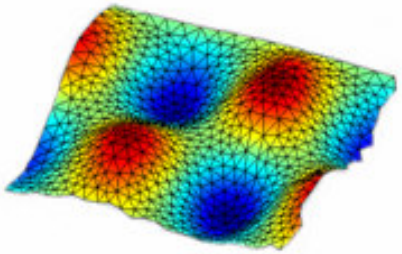








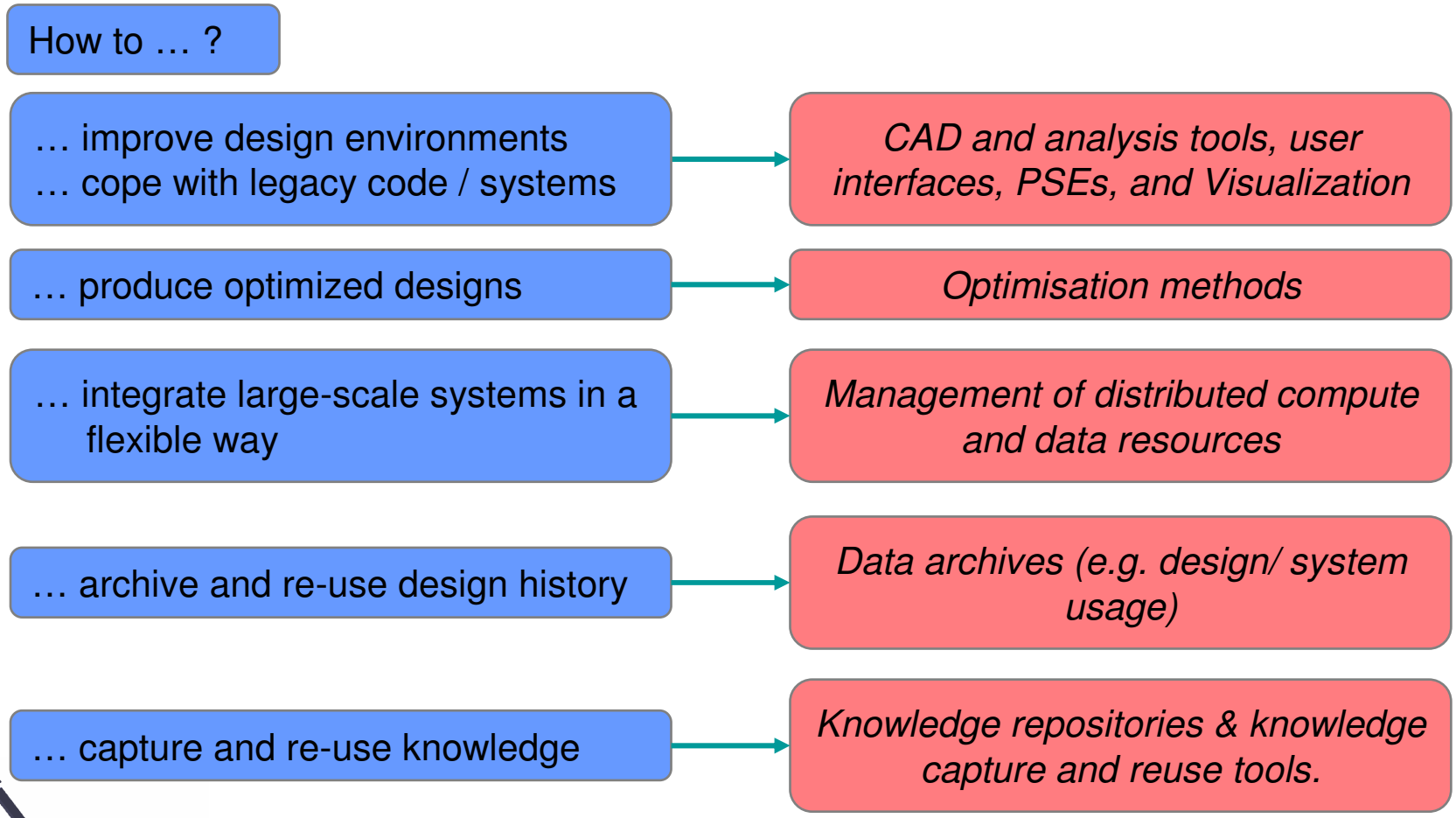




GEODISE

Design Challenges

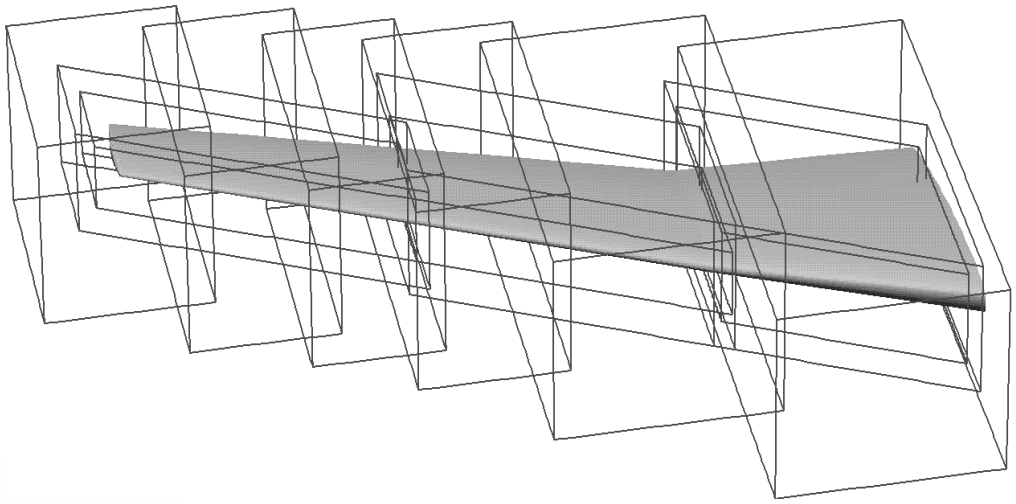
Modern engineering firms are global and distributed

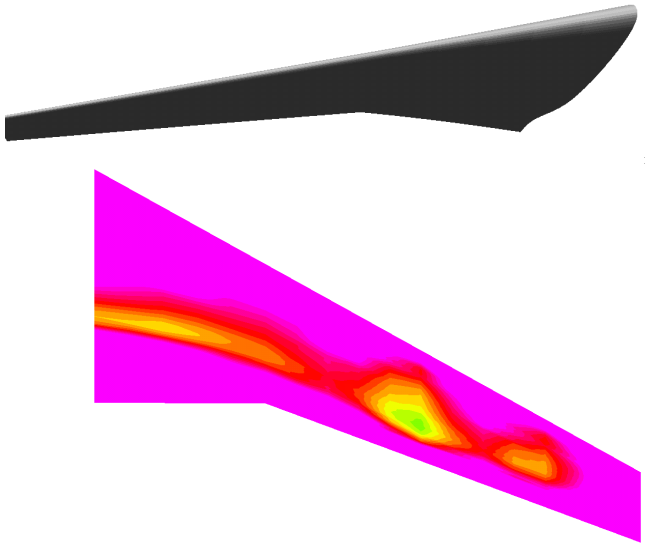


GEODISE

What is Engineering Design Search & Optimisation?

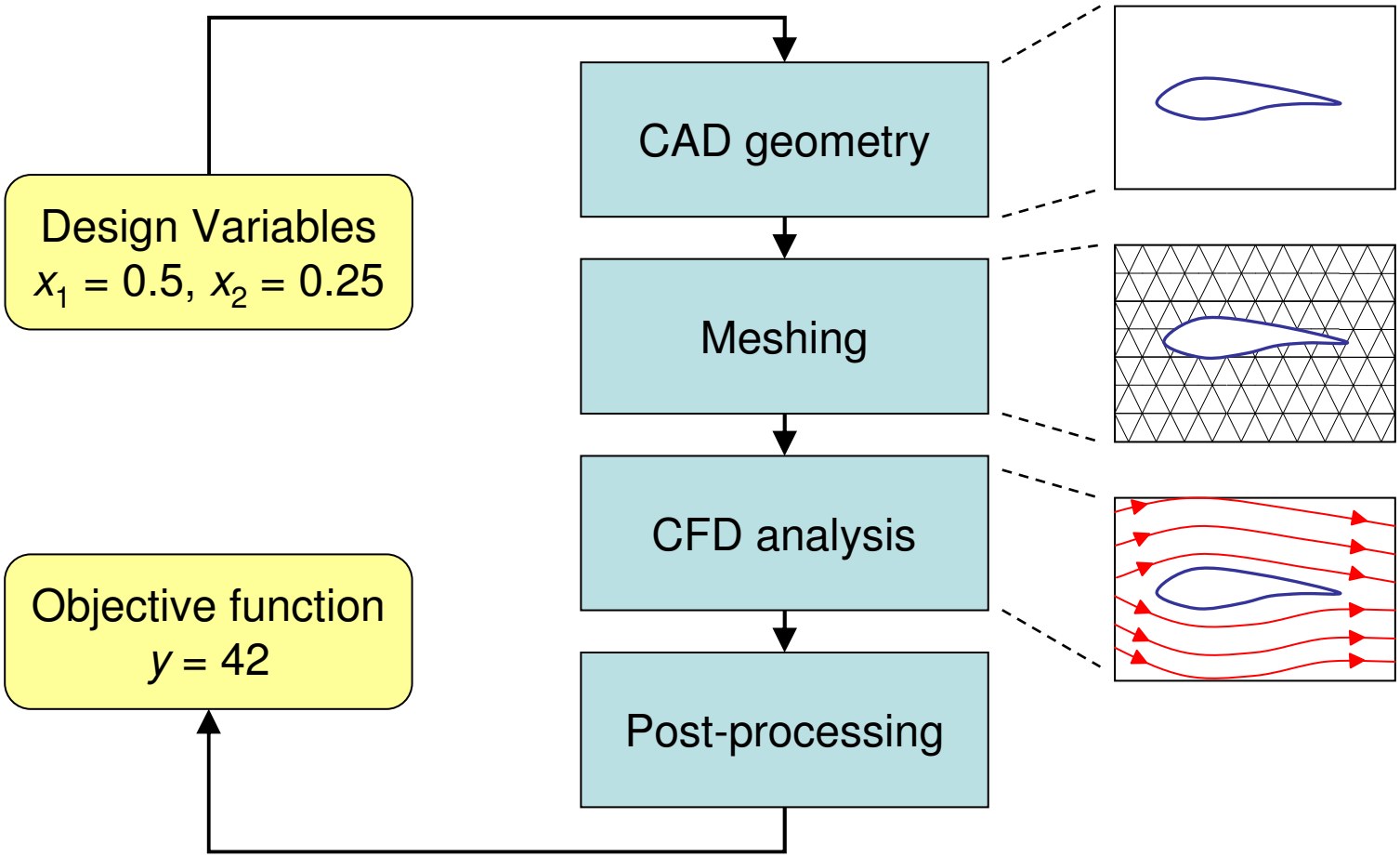
- Improving designs by exploiting engineering modelling and analysis.
- Systematically modifying variables in a design to increase, or reduce a quality measure (called the objective function), within certain constraints.
- Involves computational and data intensive processes



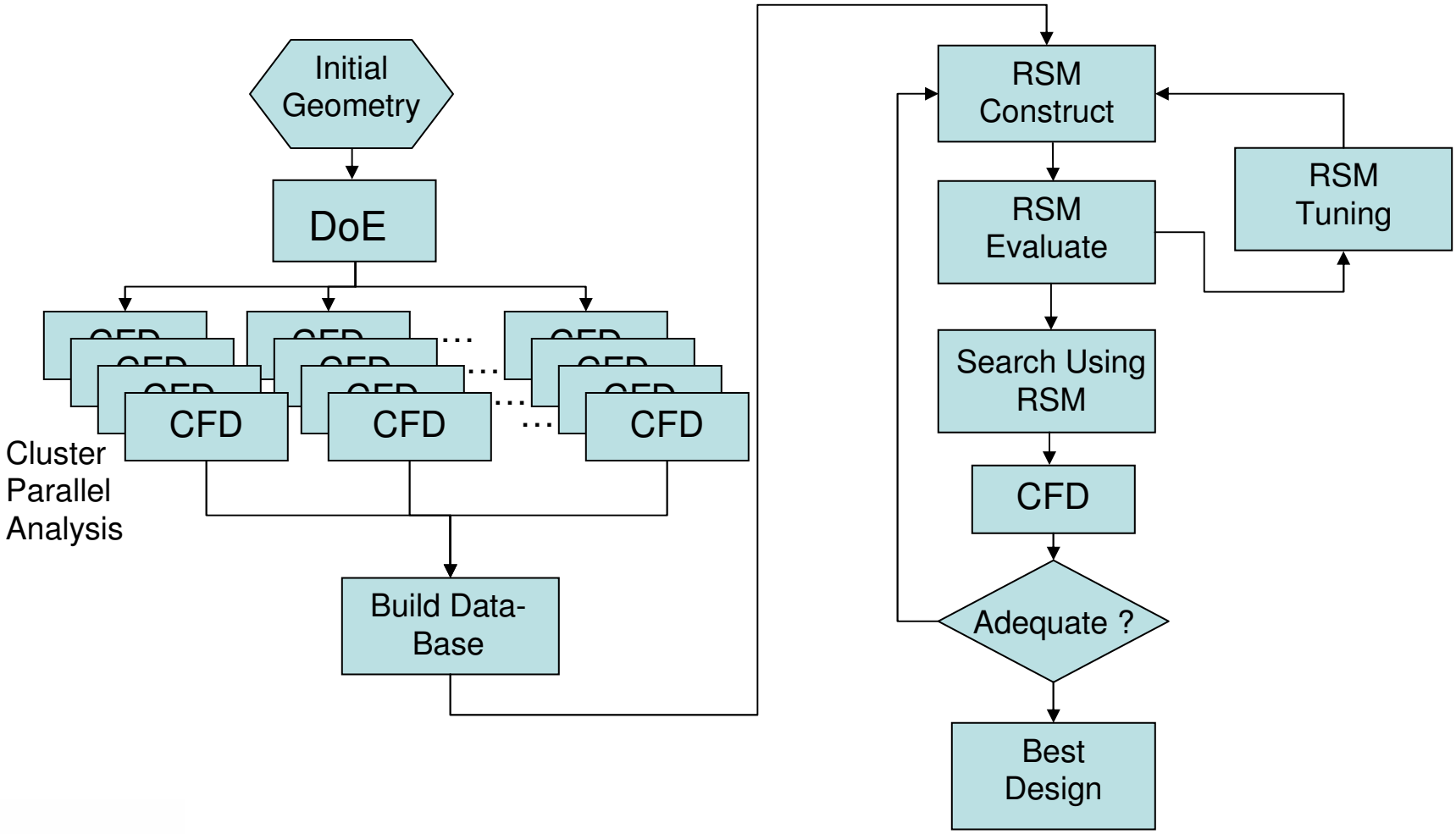


GEODISE

Defining the Objective Function

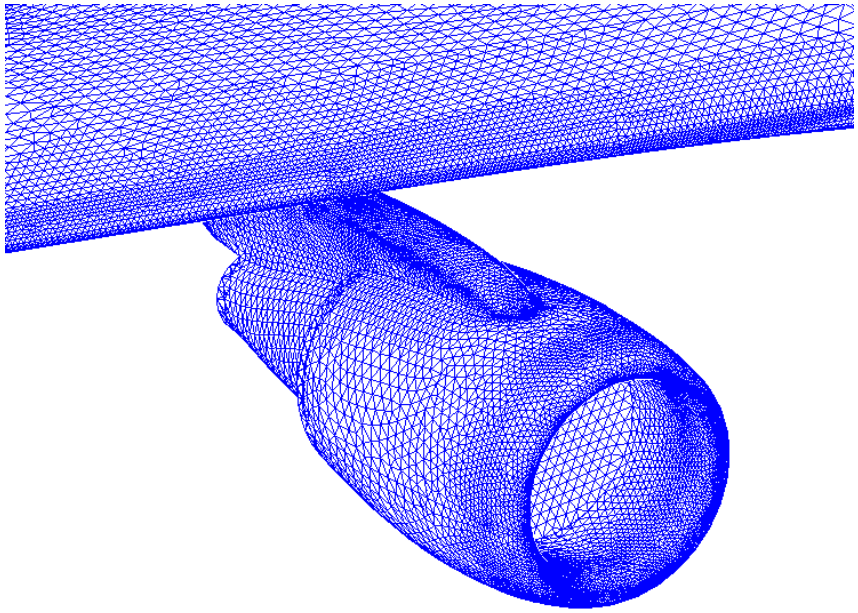


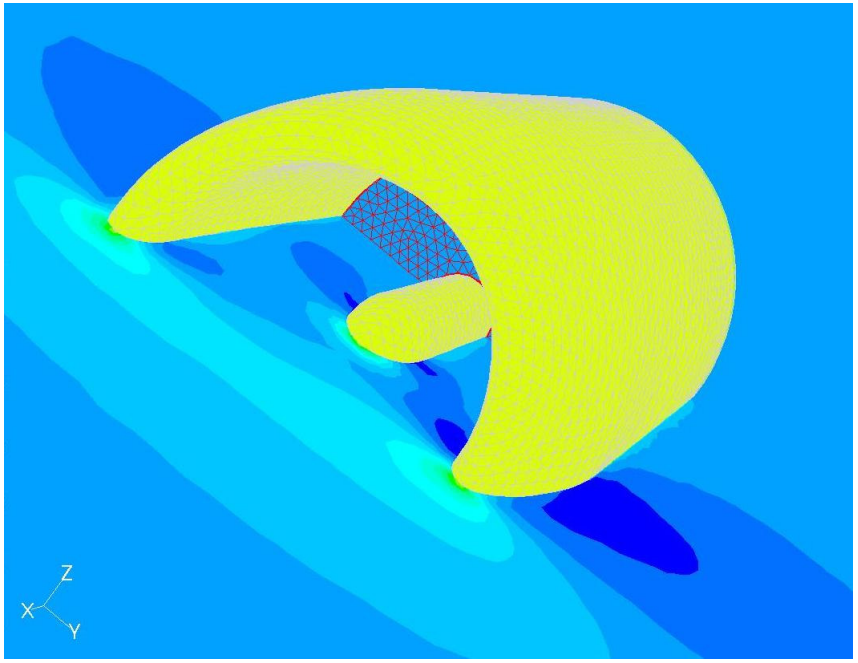
Design of Experiment & Response Surface Modelling



GEODISE

Computational Fluid Dynamics





GEODISE

Application profile – CFD / EDSO

- CFD analysis may be:
  - Computationally expensive
  - Require/produce large volumes of data
- Evaluation of an engineering objective function may require:
  - Multiple applications invoked in sequence
  - Third party proprietary applications with specific hardware requirements
  - Pre- and post- processing
  - Automation of interactive tasks into batch processes
- Optimisation algorithms may be incorporated into larger optimisation strategies
- Data reuse/analysis may prevent expensive duplicate calculations
- ***No two problems are the same***

Design Principles

Problem Solving Environments

"A PSE is a computer system that provides all the computational facilities needed to solve a target class of problems."

S. Gallopoulos, E. Houstis & J. Rice (1994)

Design Principles

- An emphasis upon the usability of the problem solving environment
- Integration of existing Grid resources and applications
- Facilitate collaboration by promoting the sharing and reuse of components, workflows, results and knowledge

Scripting languages

Why use scripting languages?

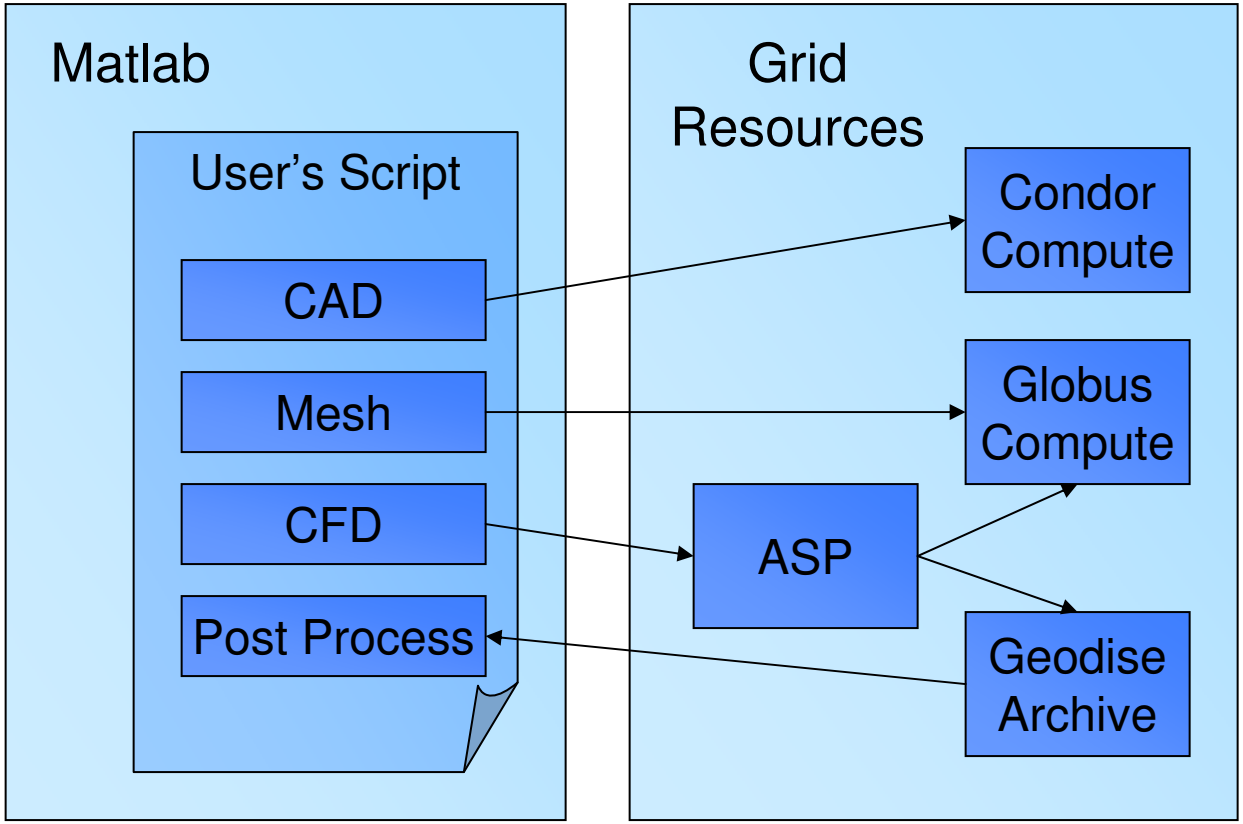
- Flexibility
- High-level functionality
- Rapid application development
- Extend the user's existing PSE
- Workflows described in a human readable format may be shared and reused

Technical Solution

Geodise Scripting Environments

- Matlab - technical computing environment
  - 500,000+ users
  - Data analysis and visualisation toolboxes
  - High-level scripting
  - Commercial product
- Jython - Java implementation of Python
  - object-oriented Python language
  - 100% pure Java
  - Active funded development
  - Open source project
- Cross platform
- Scripting complex engineering workflows

Scripting EDSO workflows

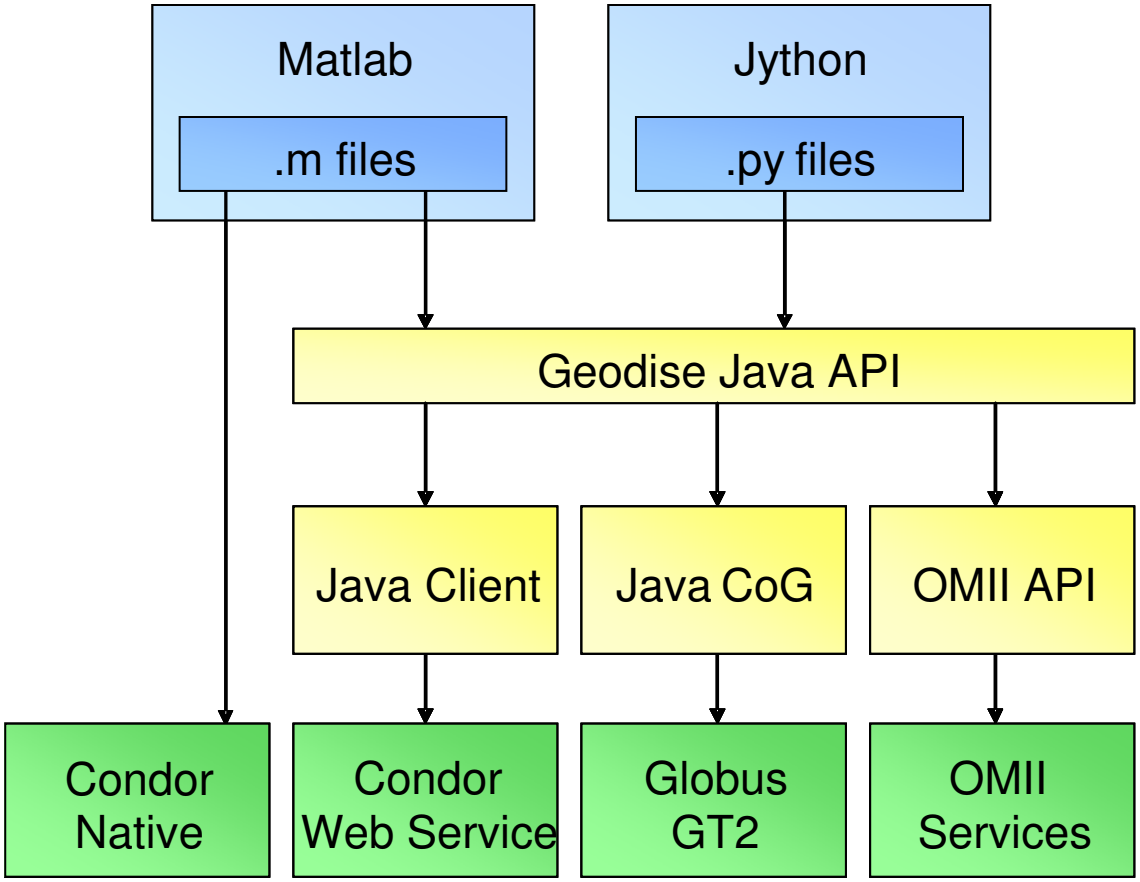


GEODISE

Technology overview

- Computational resources
  - Globus (v2.4 remains ubiquitous), via Java CoG Kit
  - Condor
  - OMII core services
  - Microsoft Compute Cluster Server 2003
  - Any others?
- XML Web Services
  - Java, Tomcat, WebSphere application server
  - ASP.NET, IIS
- Data management
  - Oracle 9i or 10g
  - GridFTP file server

Computational Toolboxes

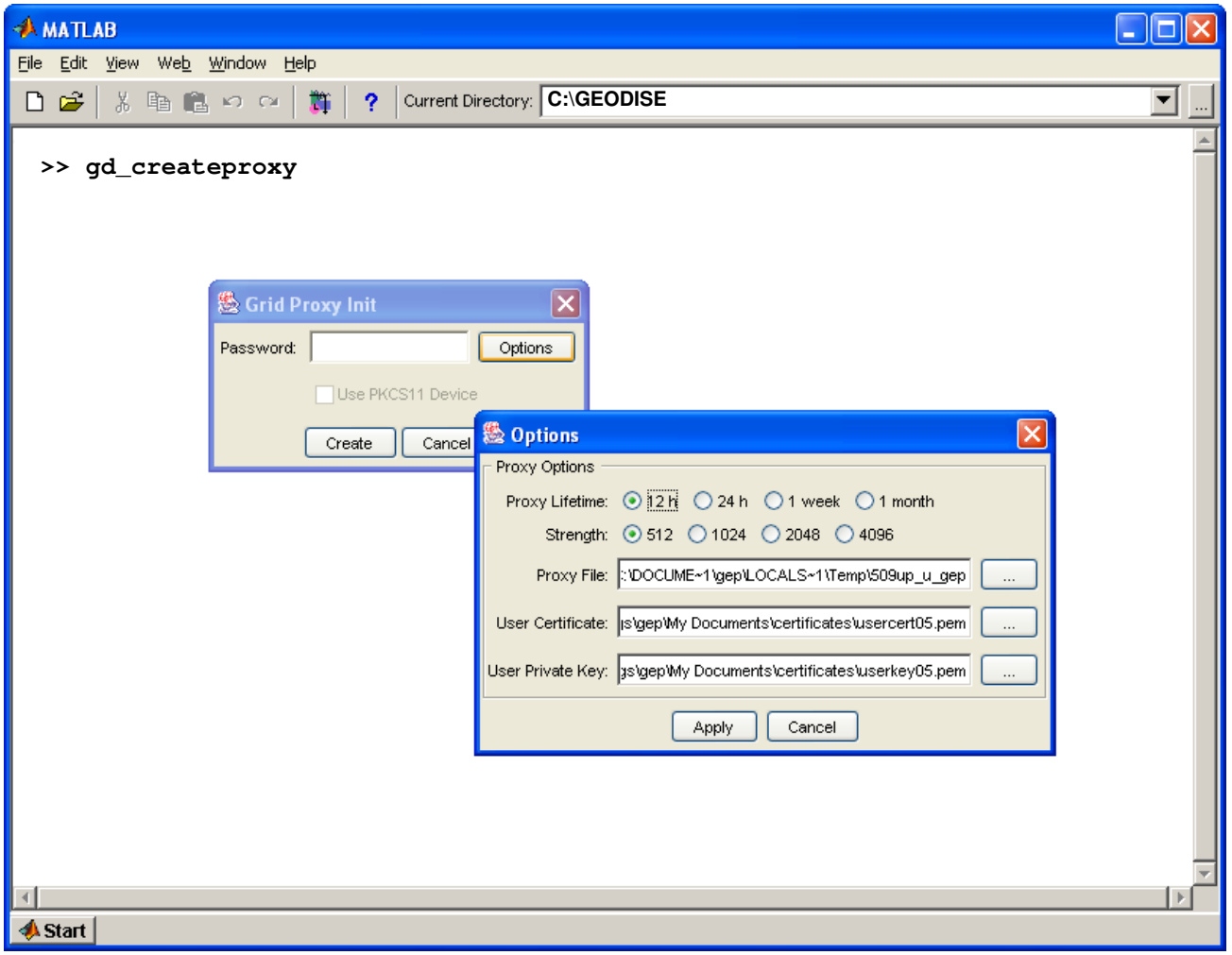


GEODISE

Certificate Management Functions

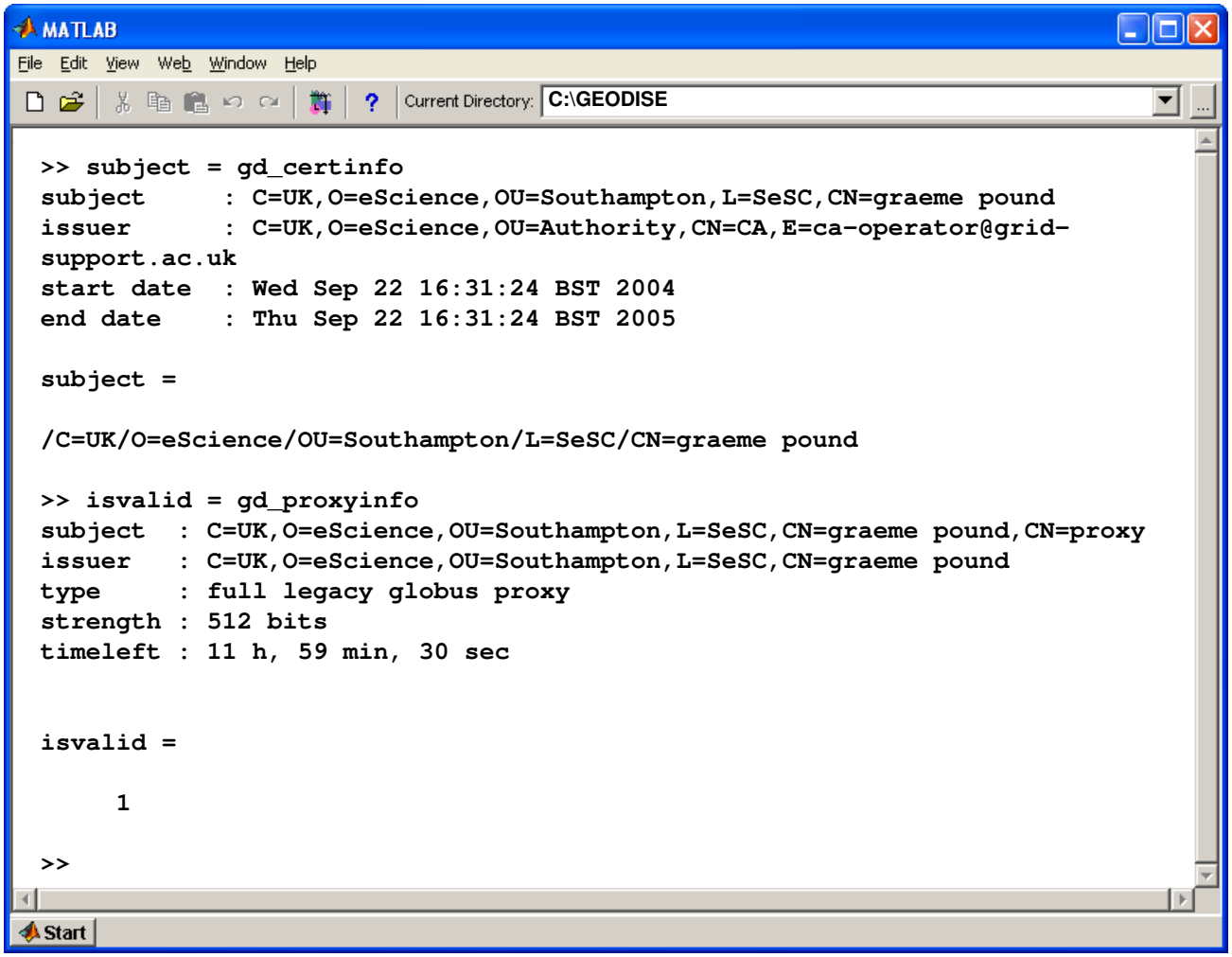
`gd_certinfo`	Returns information about the user's certificate
`gd_createproxy`	Creates a Globus proxy certificate
`gd_proxyinfo`	Returns information about the user's proxy certificate
`gd_proxyquery`	Queries whether a valid proxy certificate exists
`gd_destroyproxy`	Destroys the local copy of the user's Globus proxy certificate

Certificate Management



GEODISE

Certificate Management

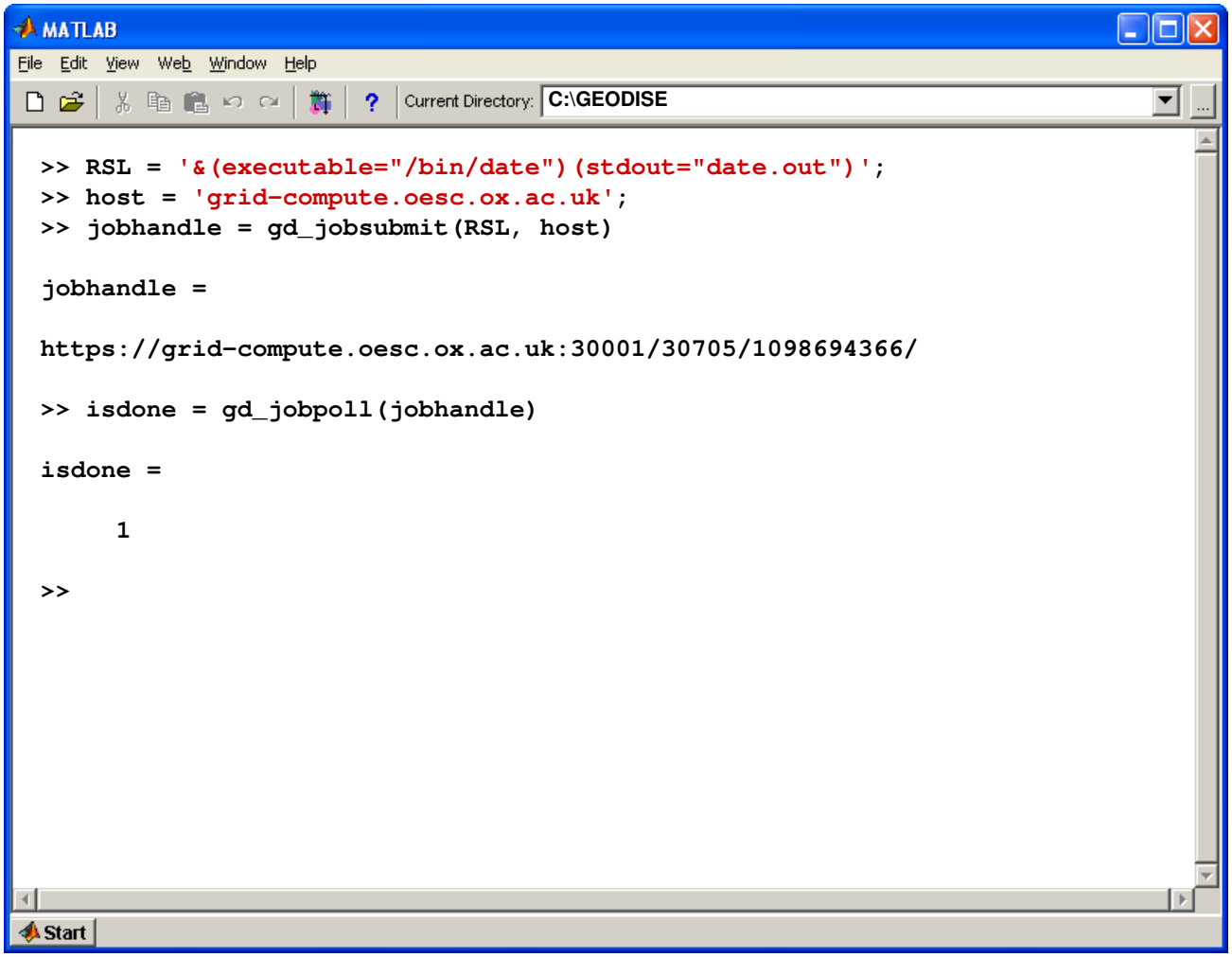


GEODISE

Job Submission Functions

`gd_jobstatus`	Gets the status of a Globus GRAM job
`gd_jobsubmit`	Submits a compute job to a Globus GRAM job manager
`gd_jobpoll`	Queries the status of a Globus GRAM job until complete
`gd_jobkill`	Kills a Globus GRAM specified by a job handle
`gd_chmod`	Changes file permissions of a file on a Globus resource
`gd_condorsubmit`	Submits a job to a Condor pool via a Globus resource
`gd_submitunique`	Submits a GRAM job to a unique working directory
`gd_listjobs`	Returns all the job handles belonging to the user

Job Submission

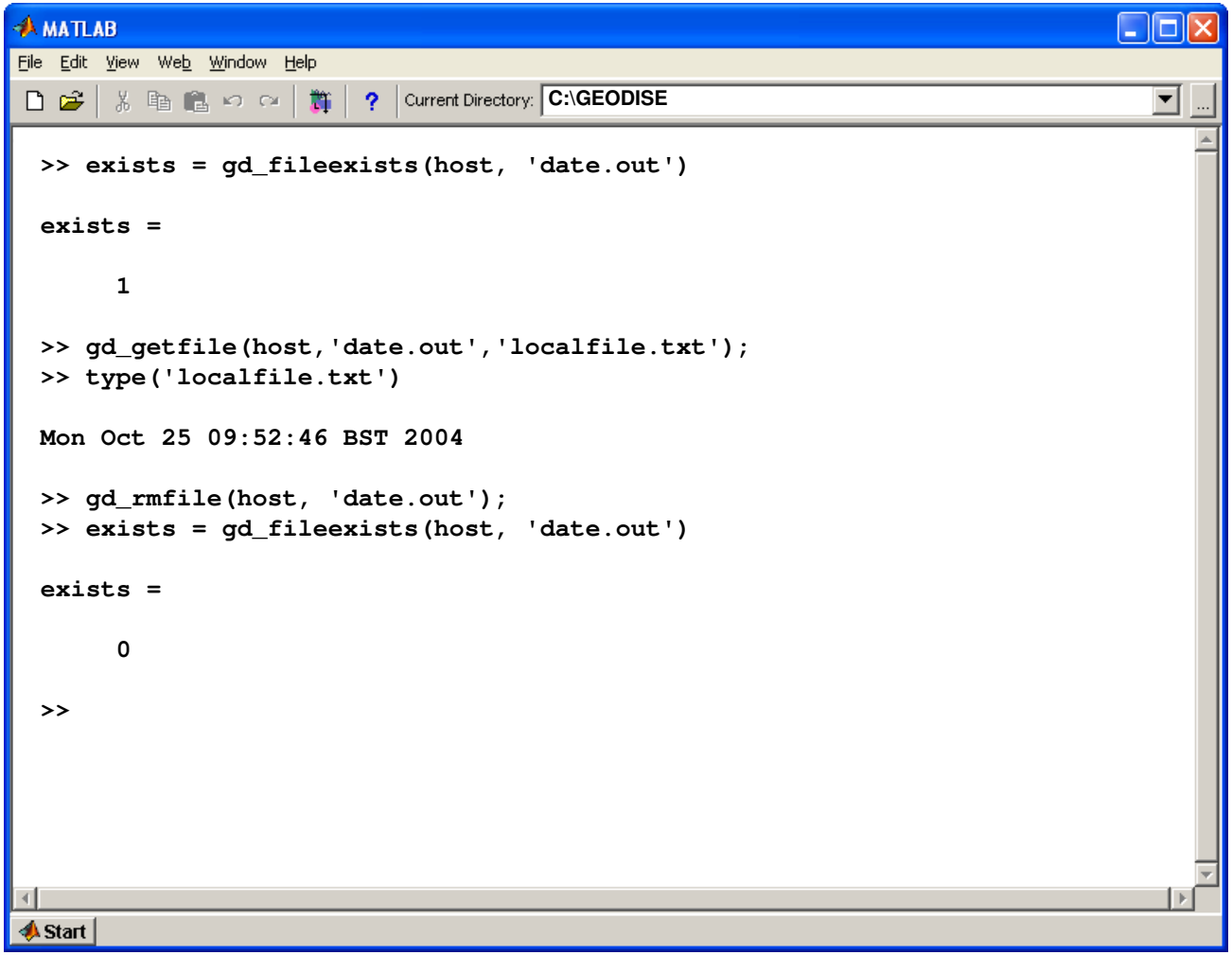


GEODISE

GridFTP File Transfer

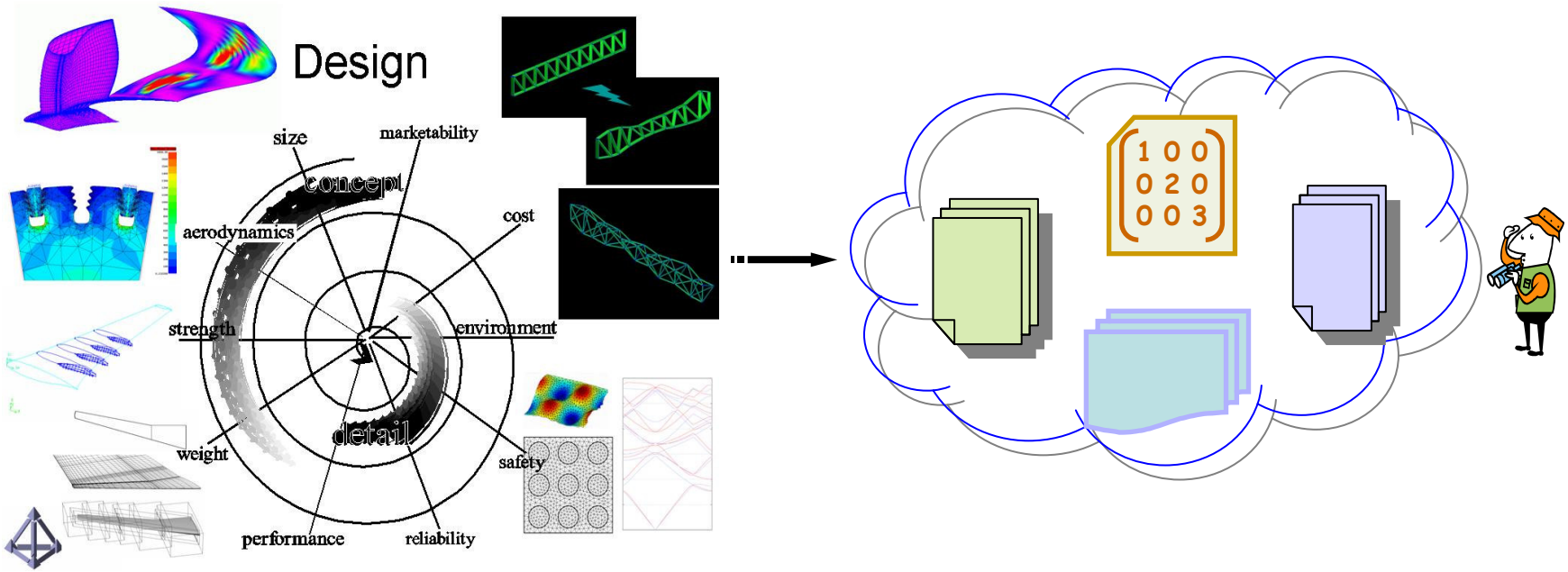
gd_getfile	Retrieves a remote file using GridFTP
gd_putfile	Puts a remote file using GridFTP
gd_transferfile	Performs a third-party file transfer using GridFTP
gd_makedir	Creates a remote directory using GridFTP
gd_listdir	Lists the contents of a directory on a GridFTP resource
gd_fileexists	Tests the existence of files on a Globus resource
gd_rmdir	Deletes a remote directory using GridFTP
gd_rmfile	Deletes a remote file using GridFTP
gd_rmuniquedir	Deletes a remote directory and its contents

File Transfer



GEODISE

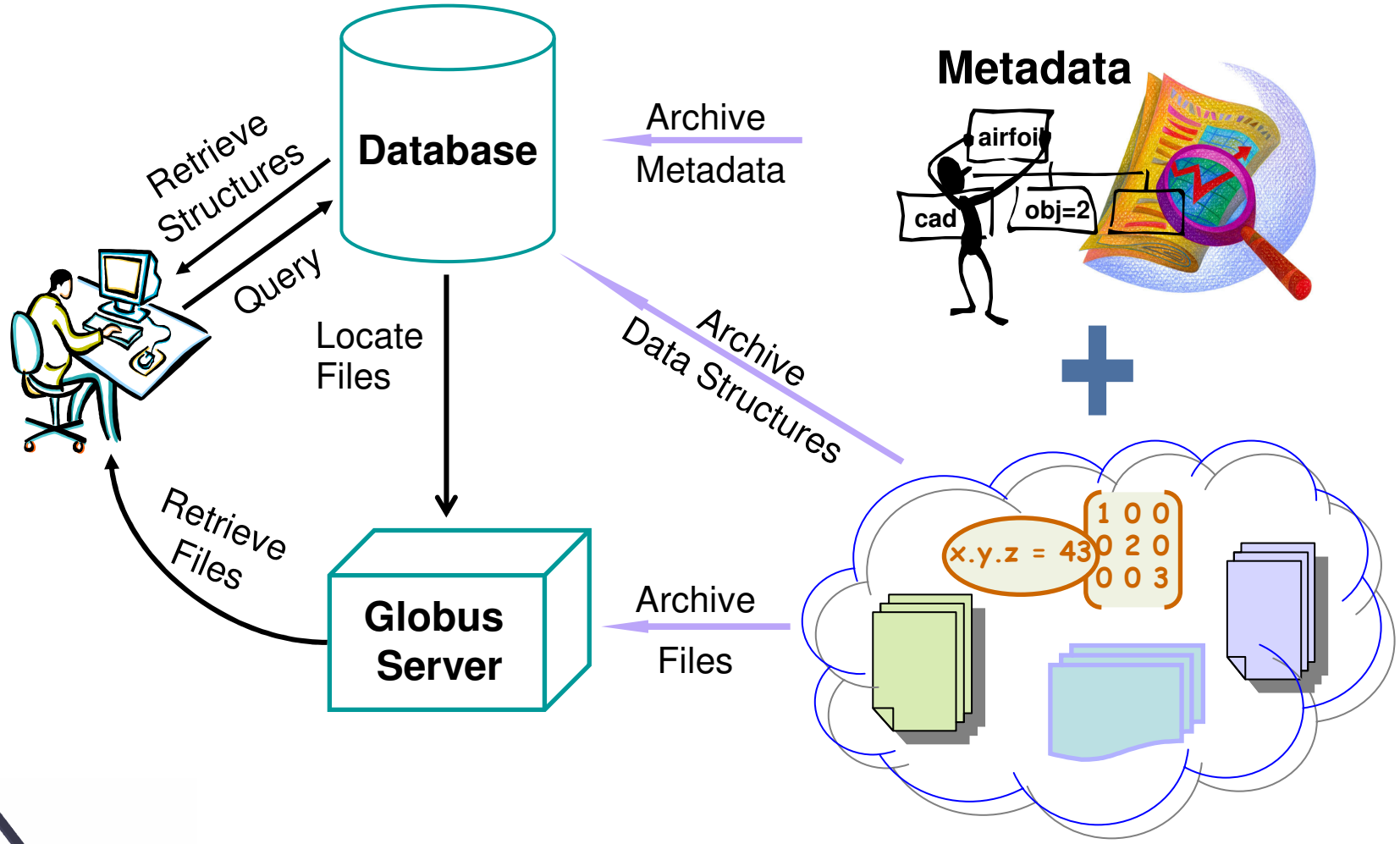
Data Management Challenges



- Data may be generated from distributed applications on the Grid.
- Traditionally data stored on file systems with little descriptive information – hard to find and share.
- Engineers may want to access the data from distributed locations.

GEODISE

Need Metadata

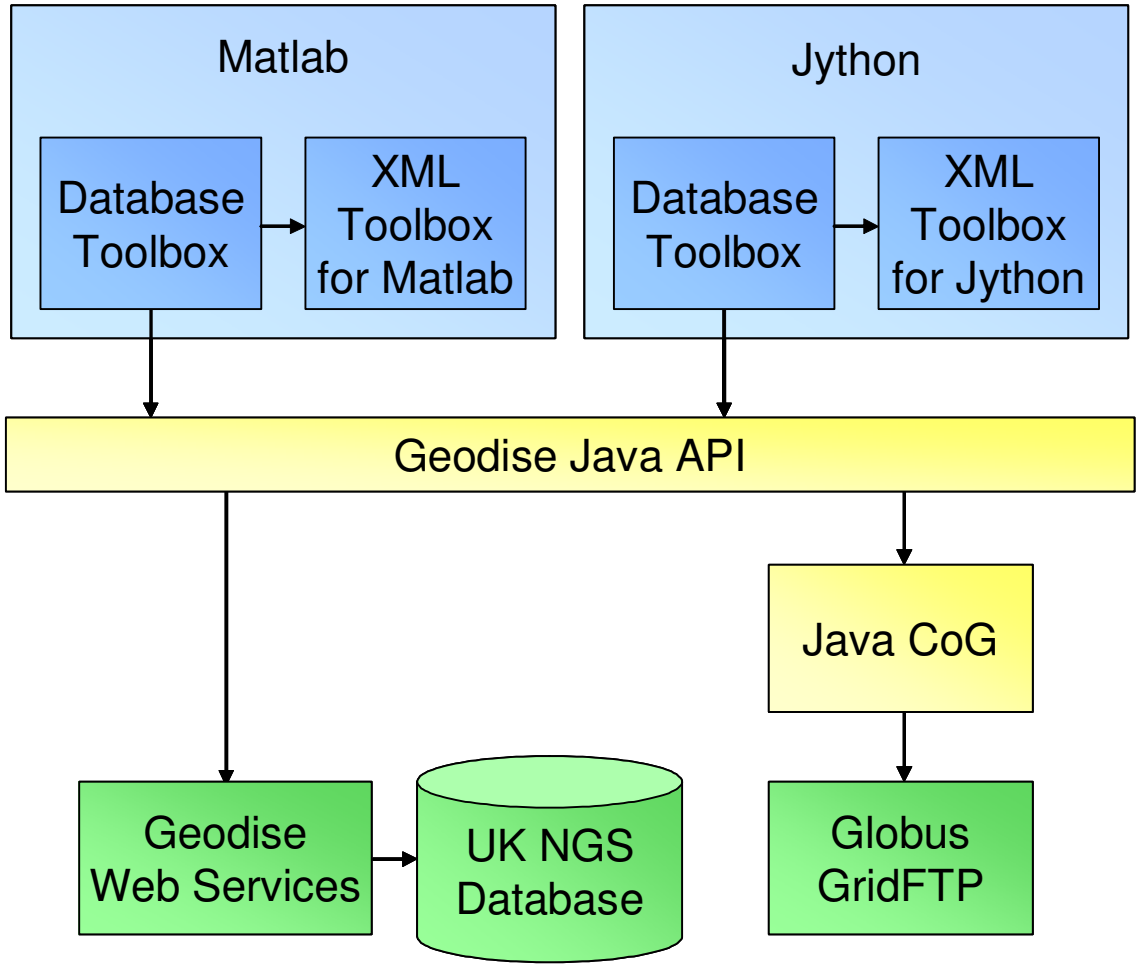


GEODISE

Database Toolbox Overview

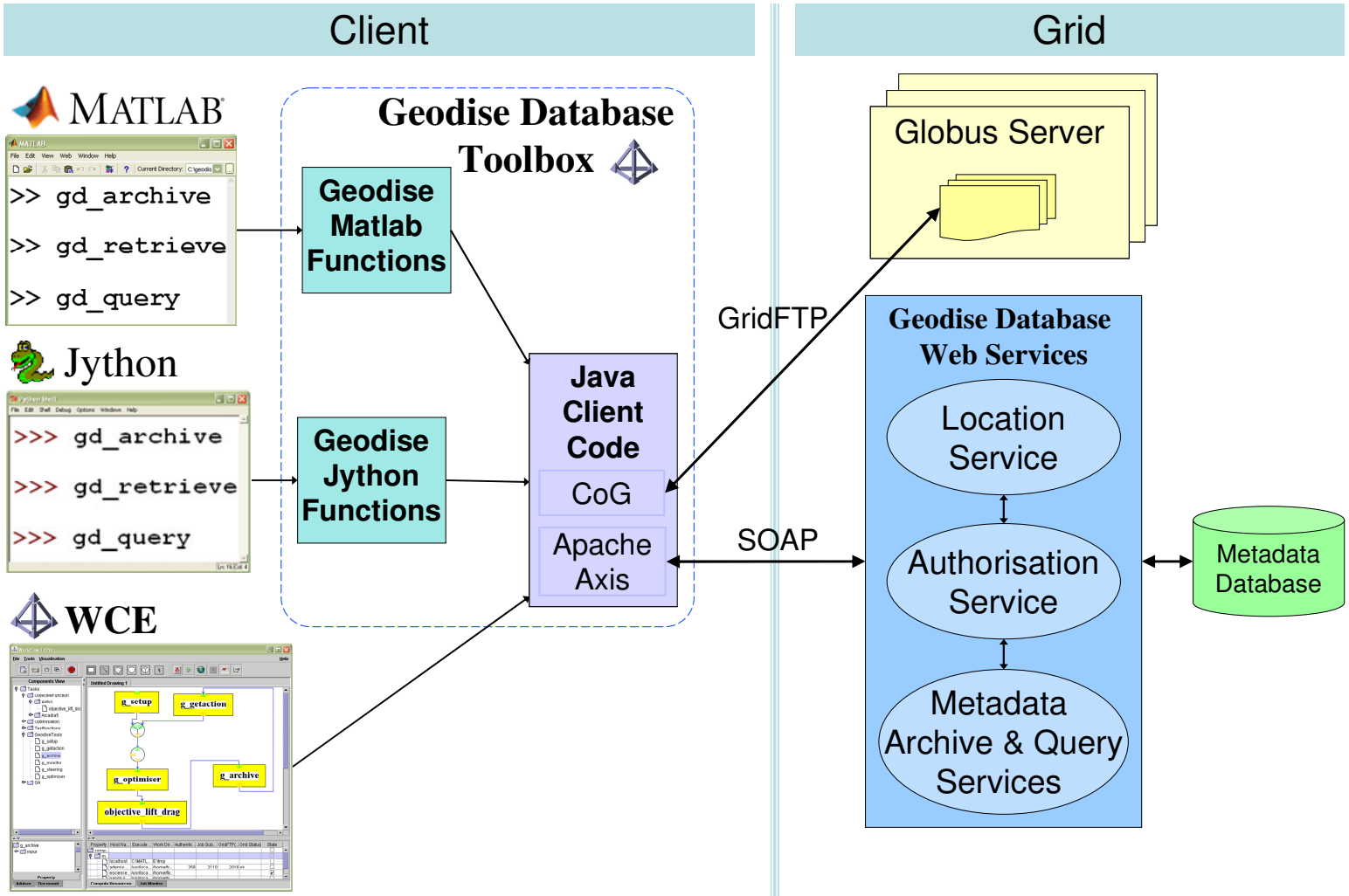
- Store data with descriptive information
  - Standard and application specific metadata.
  - Query over metadata to easily locate required data.
  - Data retrieval based on ID rather than location.
- Familiar interface for engineers
  - Wrap toolbox as Matlab/Jython functions – easy integration.
  - Can be used in Matlab/Jython scripts – popular among engineers.
- Support data aggregation (data groups) concept.
- Central and local databases (shared vs. personal).
- Secure Web service access to central database over SSL.
  - Certificate-based authentication and authorisation.

Database and XML Toolboxes



GEODISE

Architecture



GEODISE

http://www.geodise.org/

Database Toolbox

Storage service

Example:

```
%Archive data:
>> fileID = gd_archive('C:\input.dat');
%Retrieve data:
>> gd_retrieve(fileID, 'E:\tmp' )
ans = E:\tmp\input.dat
```

Metadata service

Example:

```
%Define metadata and archive file:
>> m.grids = 1;
>> m.turb_model = 'sa';
>> fileID = gd_archive('C:\input.dat', m);
```

Query service

Example:

```
>> r = gd_query('standard.userID = me & grids < 2');
>> gd_display(r):
standard.userID = me
standard.ID = input_dat_8a184899-ad2d-4055-aad9-a1
grids = 1
```

XML Toolbox

- Serialise/Deserialise Matlab variables to and from XML
- Read any XML document into a Matlab structure
- High level functions (**`xml_save()`** and **`xml_load()`**)
- 1566 downloads from MatlabCentral in 18 months
- User base in academic & commercial research

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

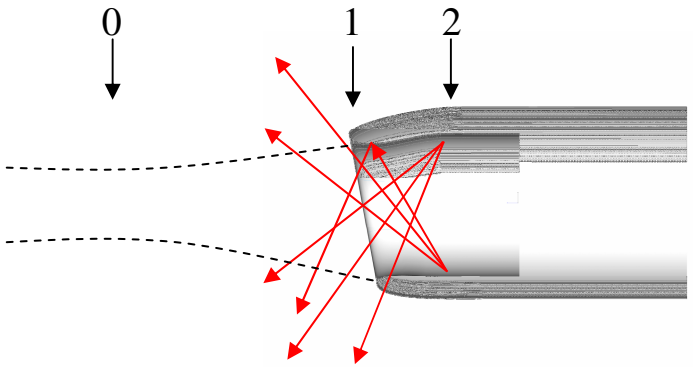
```
>>> from gdxml import xml_load
>>> v = xml_load( 'demo.xml' )
>>> print v
{'b': 3.1415926535897, 'c': 'a character string'}
```

OptionsMatlab

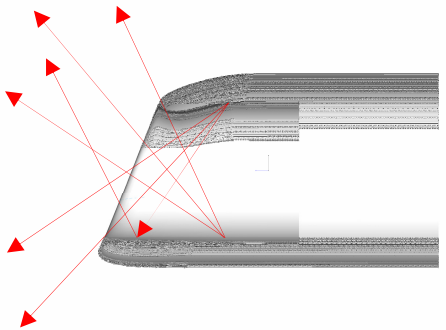
- Matlab interface to the Options design exploration system
- Reduce barriers to entry
- *State of the art* design search and optimisation algorithms
- User's objective and constraint functions exposed as Matlab functions
- Grid-enabled job brokers easily incorporated
- Composition of complex optimisation strategies

Application Examples

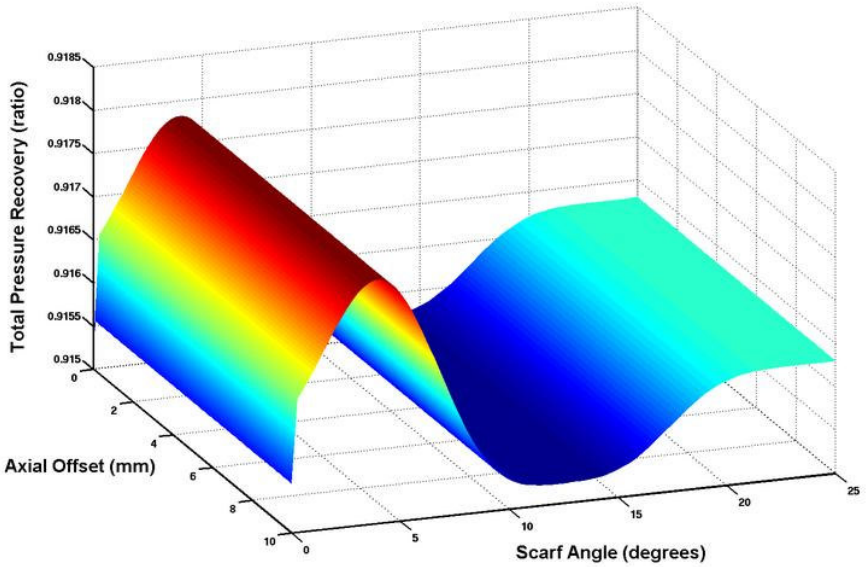
CFD 3D engine nacelle optimisation



Conventional Inlet


Negative Scarf Inlet

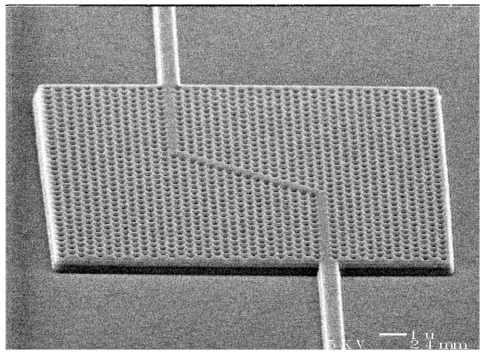
- Goal - reduce ground noise generated by fan when plane takes off.
- Optimise aerodynamic performance when scarf angle is varied.



GEODISE

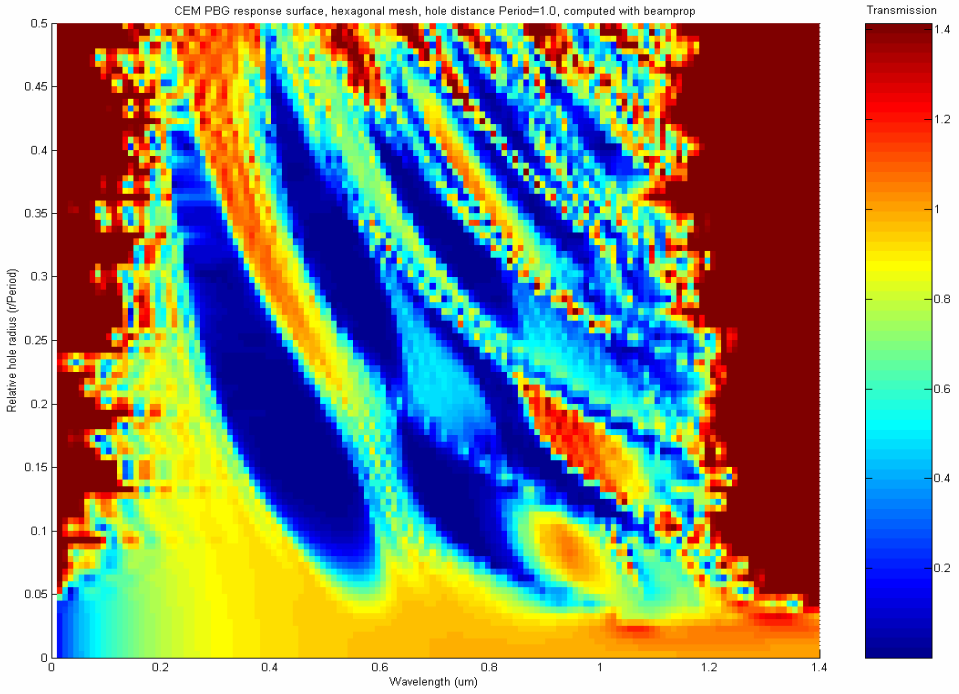
http://www.geodise.org/

GEM: Electromagnetic optimisation



Transmission of light properties through a photonic crystal.

Parameters are radius of holes and light wavelength.

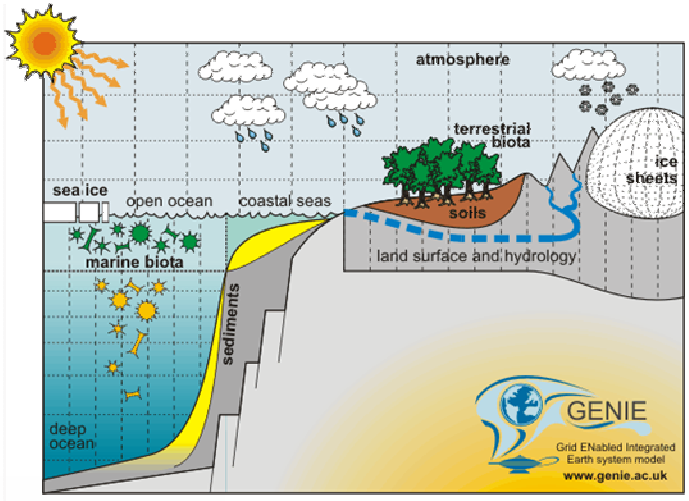


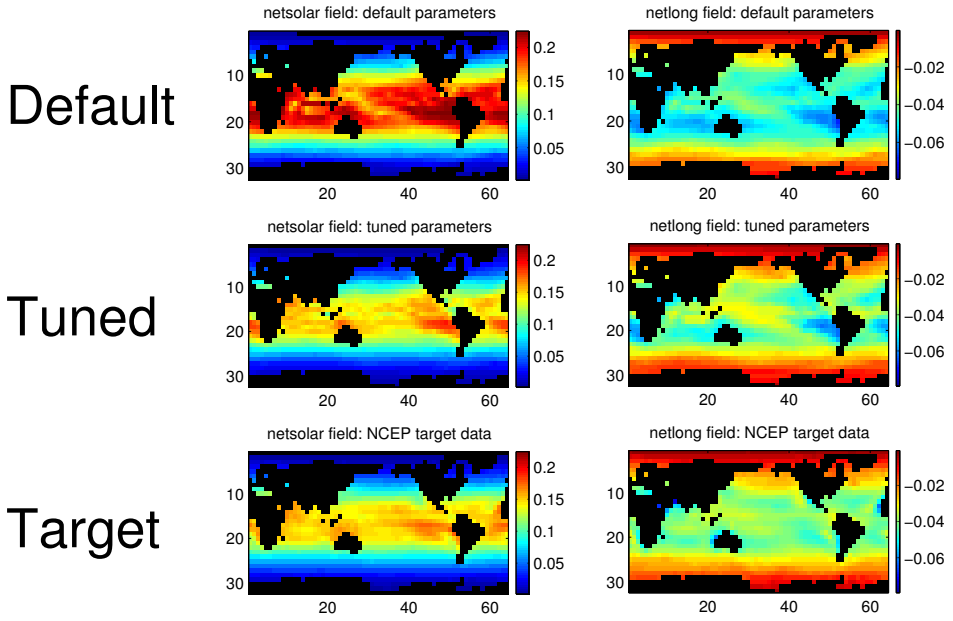
- Large number of designs, parameters and solutions.
- Query for a particular data range to post-process.

GEODISE

GENIE: Earth System Modelling

- GENIE supports environmental scientists modelling long term climate change.
- Tuning Earth system model components
- Optimising a parameterised model over a multi-dimensional state space.
- Database used for monitoring, sharing, post-processing and fault tolerance.

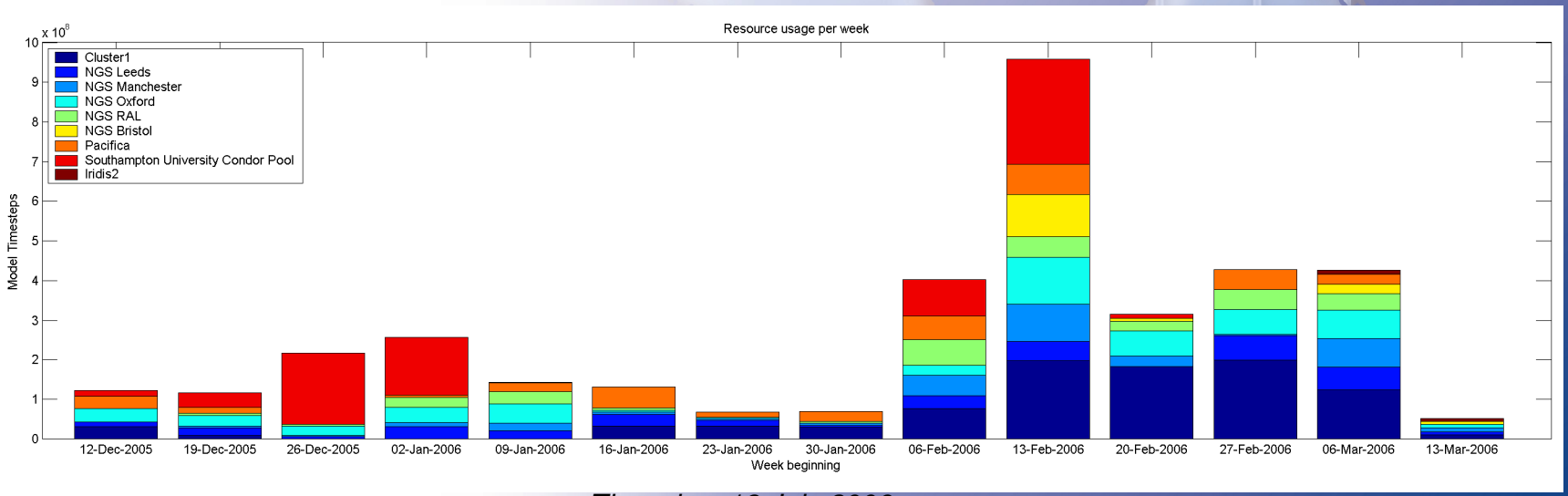




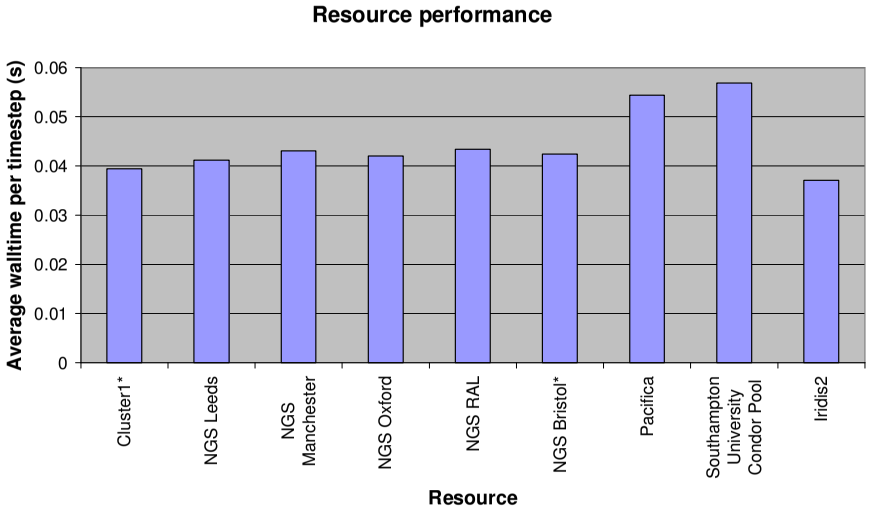
GEODISE

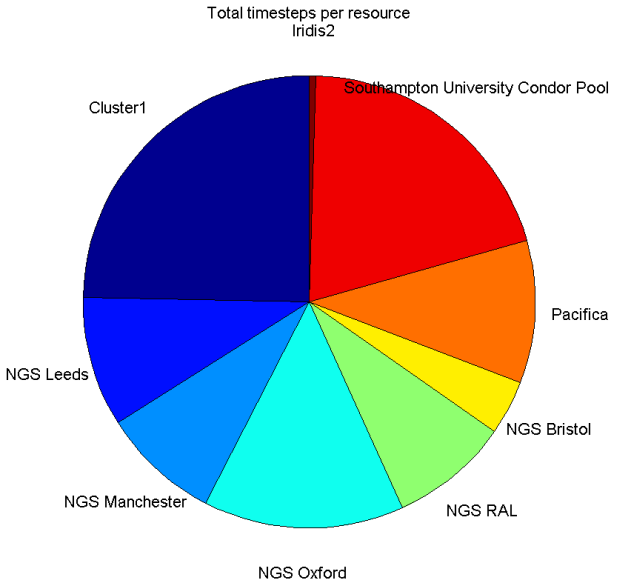
Resource Usage

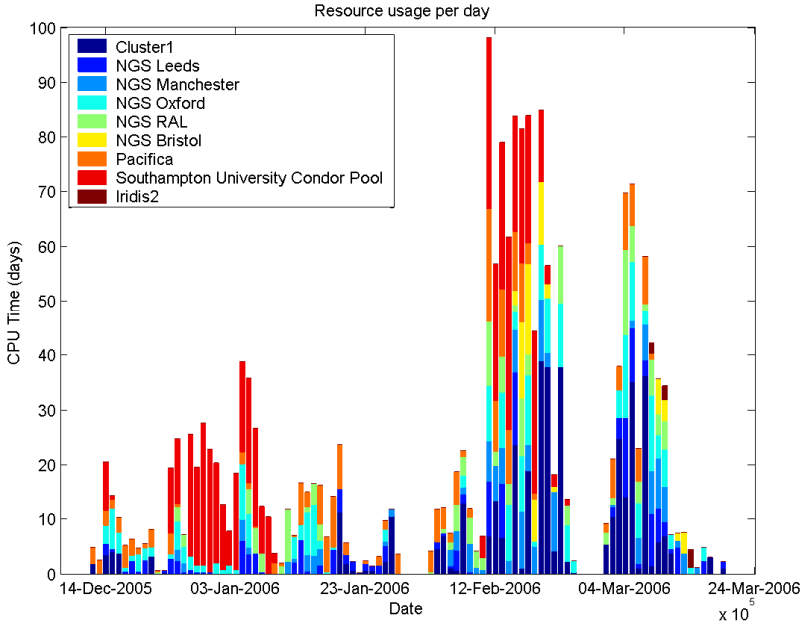
- 5 client installations
- 9 Grid resources exploited
- 352 simulations defined (1000 and 2000 yrs)
- 3,736 compute tasks submitted
- 46,992 CPU hours (estimated)
- 428,000 IGCM-GOLDSTEIN years performed

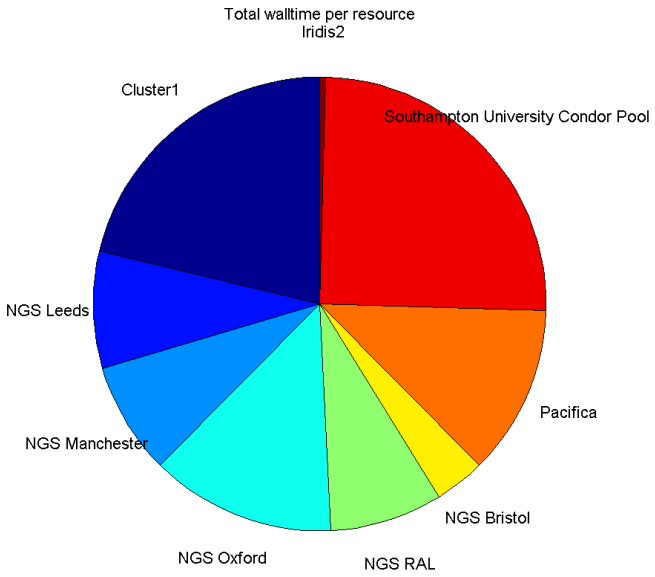


Resource Usage









Conclusions

- Many alternative possible approaches
- You should consider:
  - Profile of the end user
  - User requirements
  - Potential modes of use
  - Available resources
- User feedback essential for development
  - User experiences may differ from your preconceptions