

# Computation services: APIs and Parallel Jobs

Richard Hopkins



# Policy for re-use



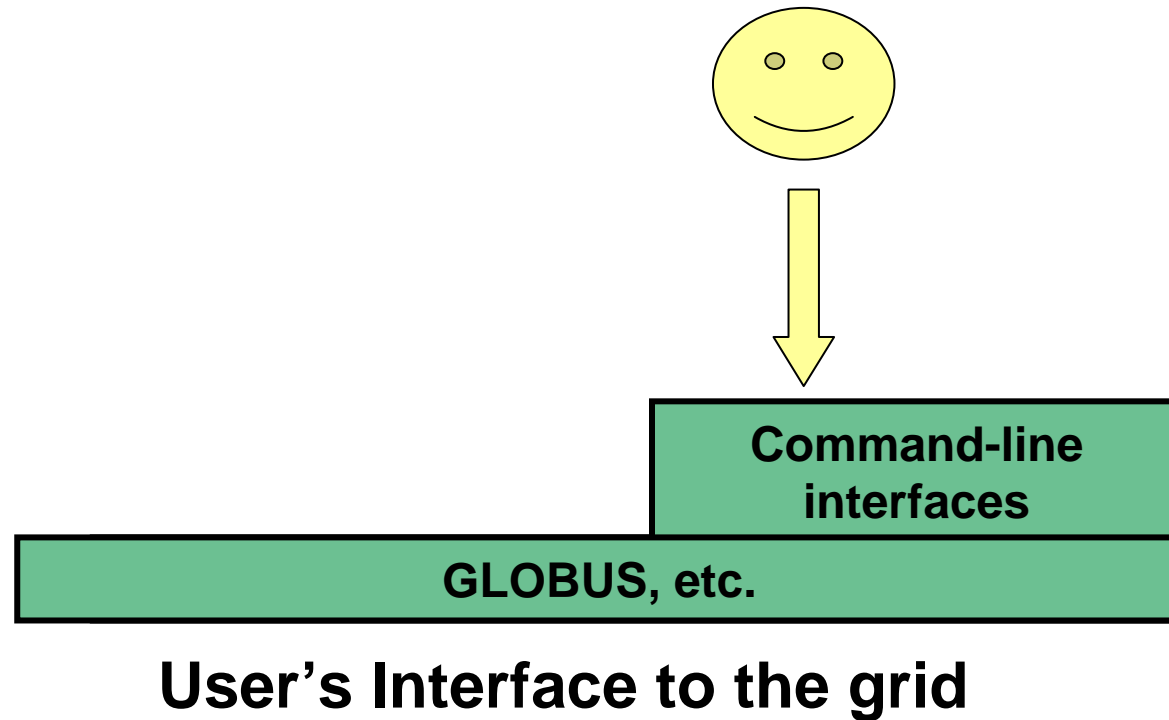
- This presentation can be re-used, in part or in whole, provided its sources are acknowledged.
- However if you re-use a substantial part of this presentation please inform [training-support@nesc.ac.uk](mailto:training-support@nesc.ac.uk). We need to gather statistics of re-use: number of events and number of people trained. Thank you!!

# Overview

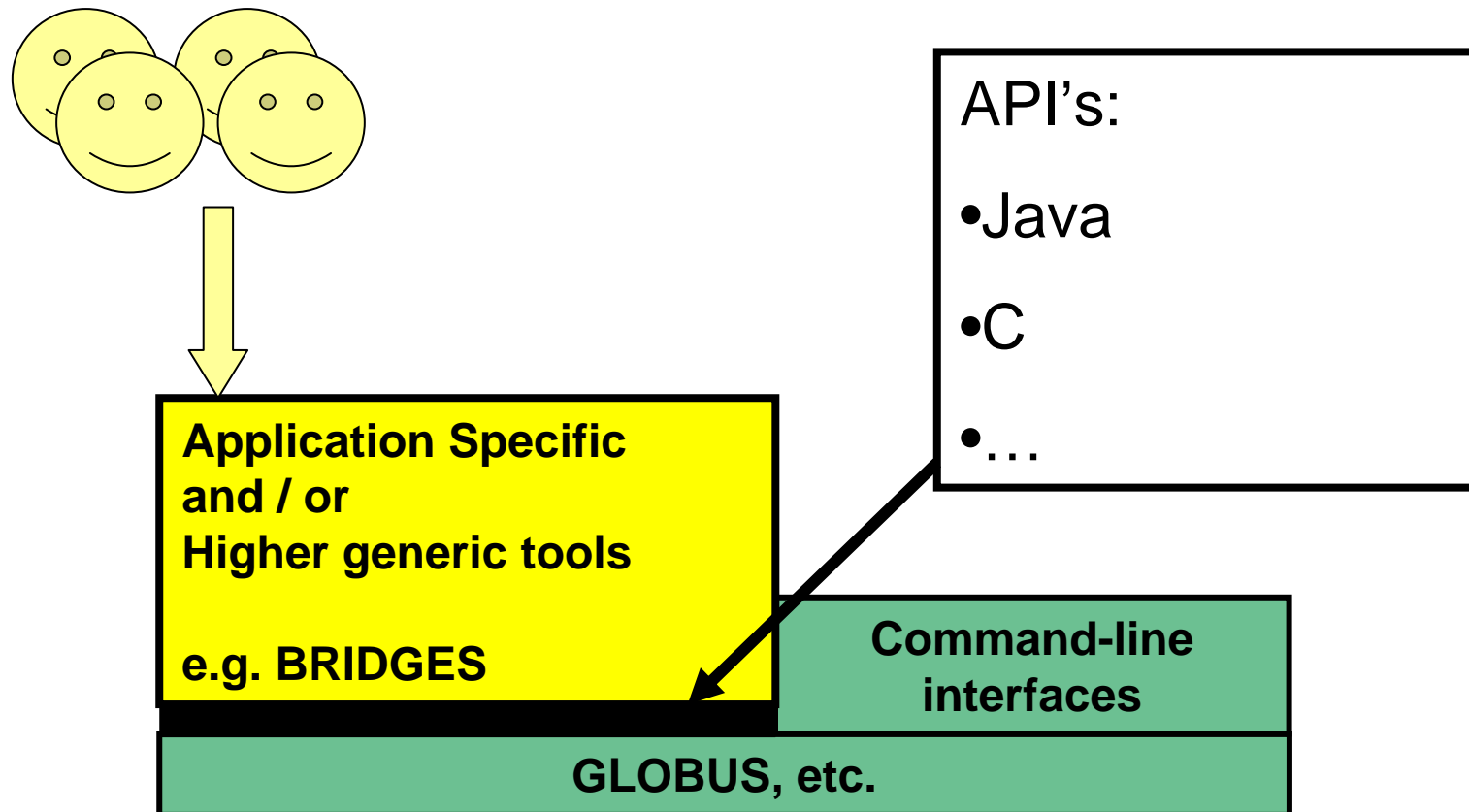


- The C and Java API's to the low-level tools
- Using multiple processors

# Job submission so far

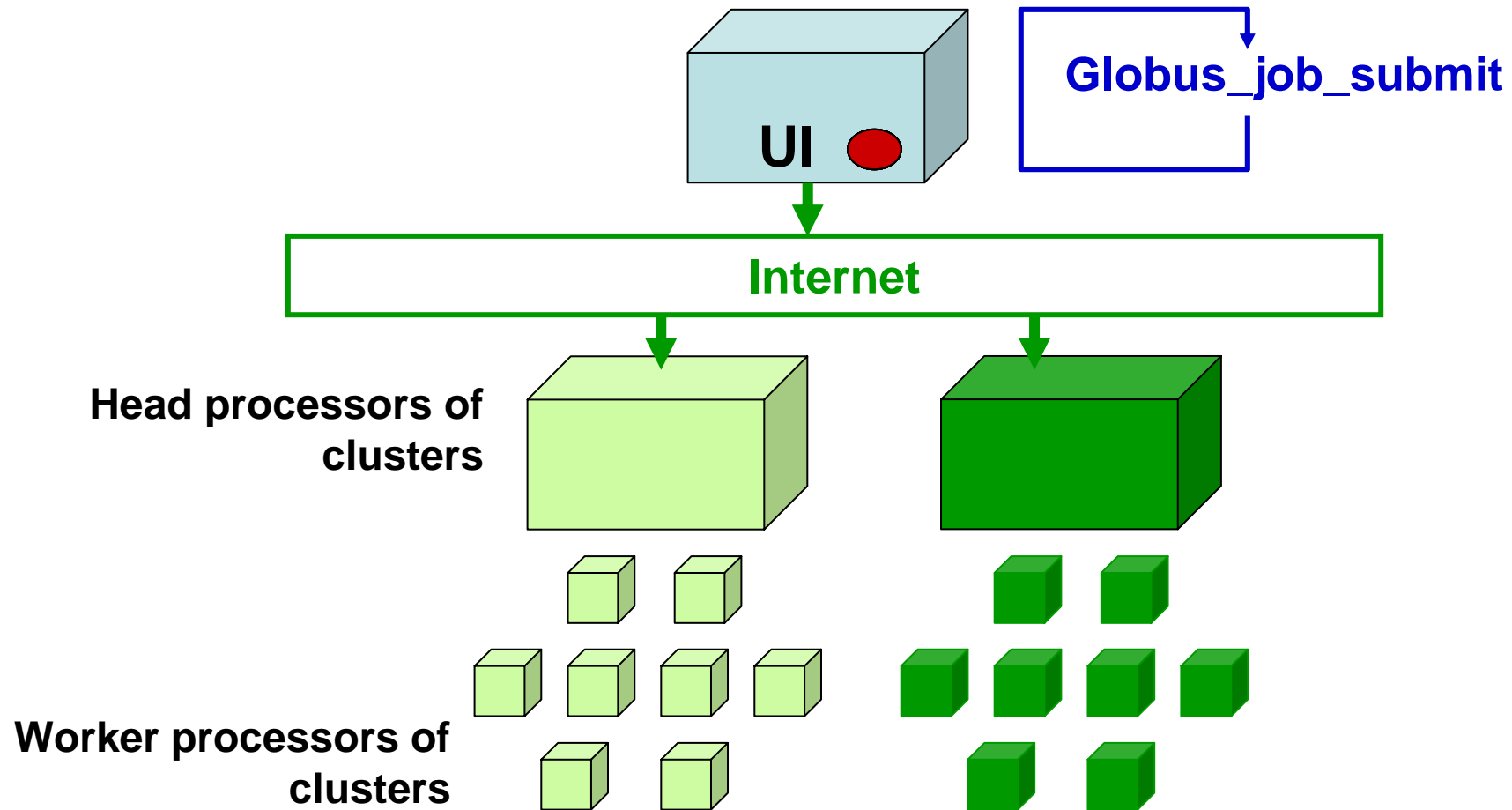


# Application-specific tools



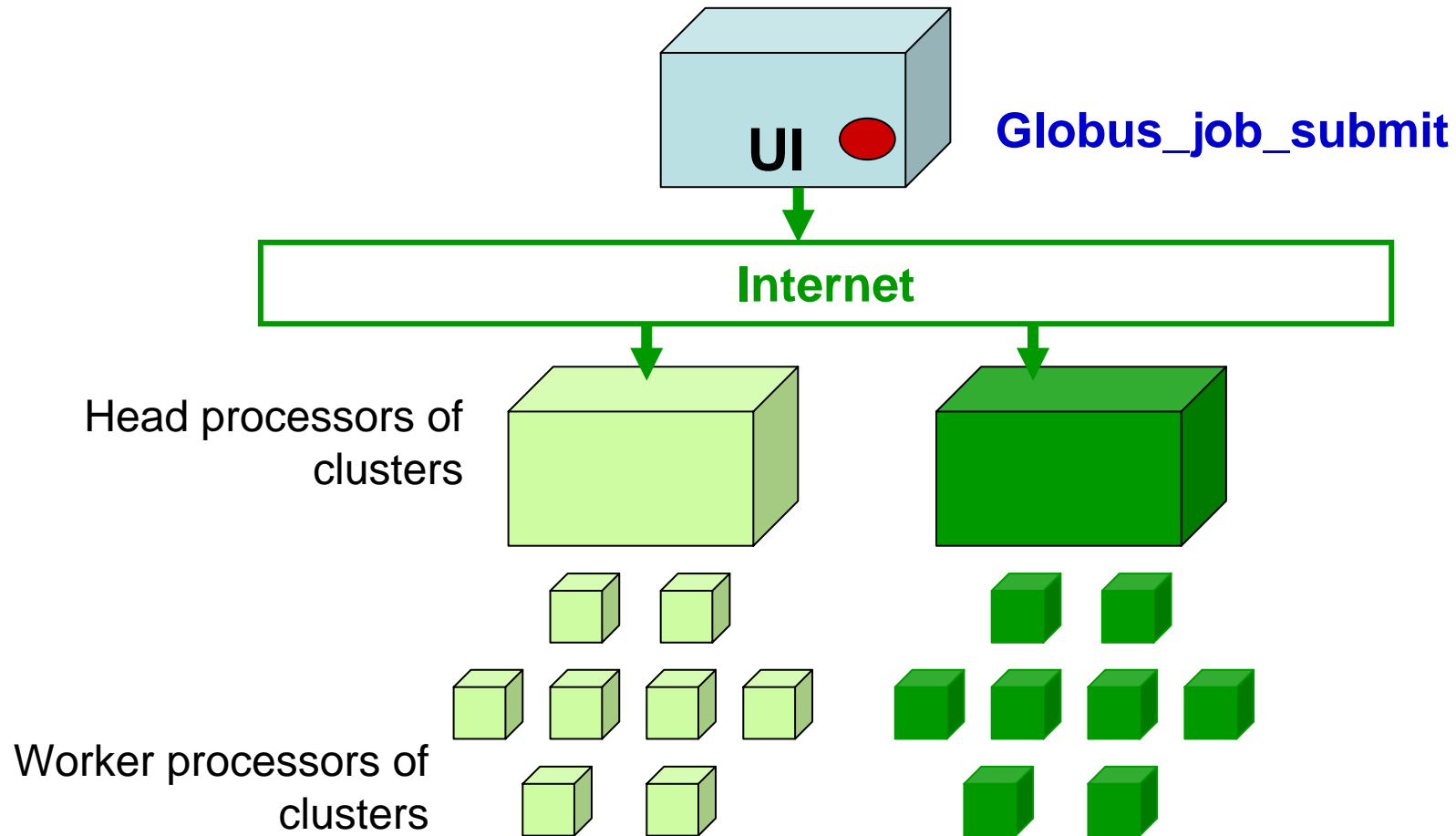
**User's Interface to the grid**

- C <http://www.globus.org/developer/api-reference.html>
- “Community Grid” CoG <http://www.cogkit.org/>
  - Java, Python, Matlab
  - (very limited functionality on Windows – no GSI)



Processes run without any communication between them

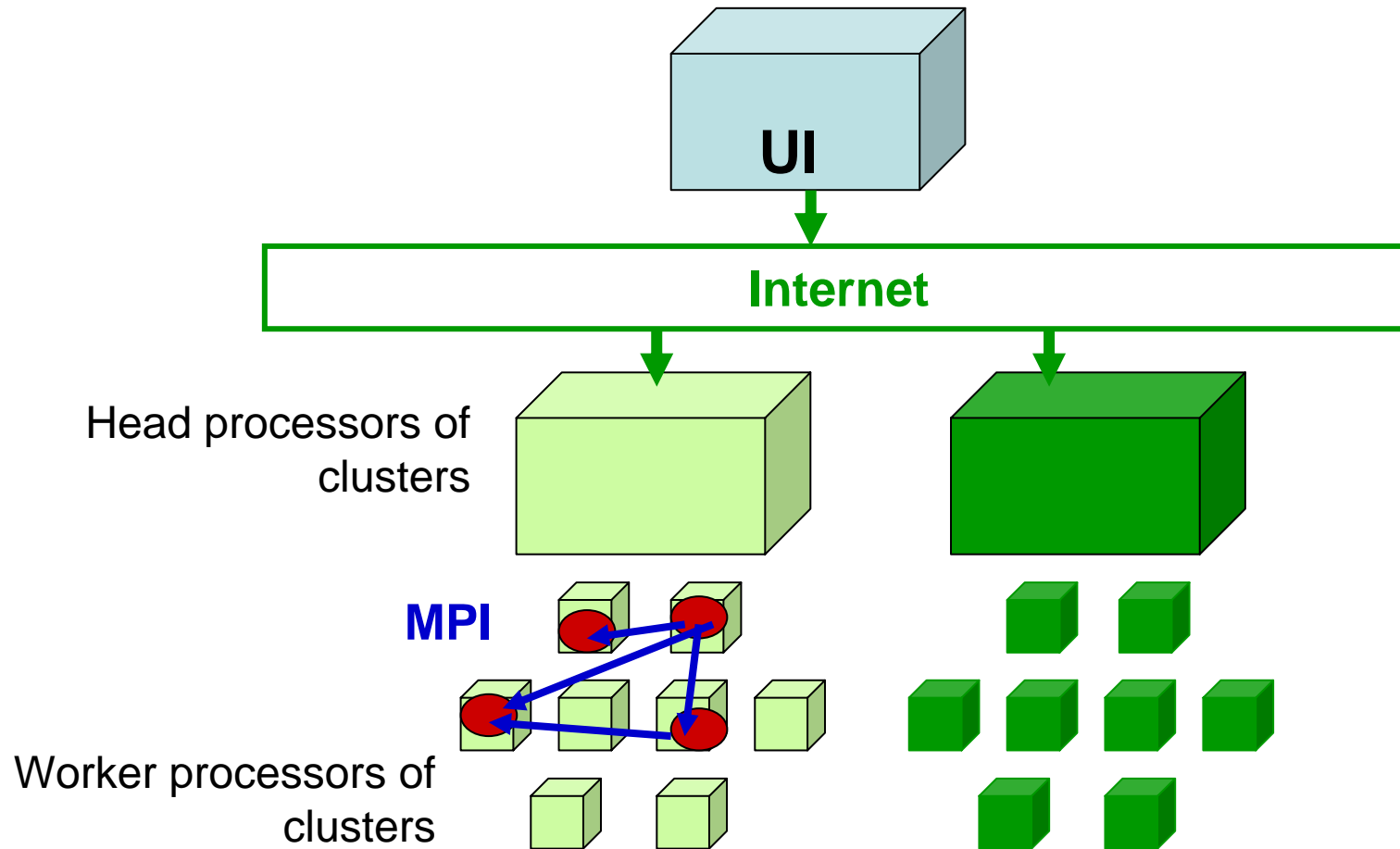
# Communicating Processes



Processes send messages to each other – Must run on same cluster



# Communicating Processes



Processes send messages to each other – Must run on same cluster

- How could the task be split into sub-tasks?
  - By functions that could run in parallel??!
  - By sending different subsets of data to different processes?  
More usual ! Overheads of scatter and gather
- Need to design and code carefully: **be alert to**
  - sequential parts of your program (if half your runtime is sequential, speedup will never be more than 2)
  - how load can be balanced (64 processes with 65 tasks will achieve no speedup over 33 processes)
  - Deadlock!
- MPI functions are usually invoked from C, Fortran programs, but also Java
- Several example patterns are given in the practical.  
Many MPI tutorials are on the Web!

1. C API Example
  2. Java API usage
  3. Concurrent processing – from Java
- Follow link from agenda page
  - [http://homepages.nesc.ac.uk/~gcw/NWGrid/GRAM\\_2.html/](http://homepages.nesc.ac.uk/~gcw/NWGrid/GRAM_2.html/)