NGS

National Grid Service

# NGS computation services: APIs and Parallel Jobs

# 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. We need to gather statistics of re-use: number of events and number of people trained. Thank you!!

NGS
National Grid Service

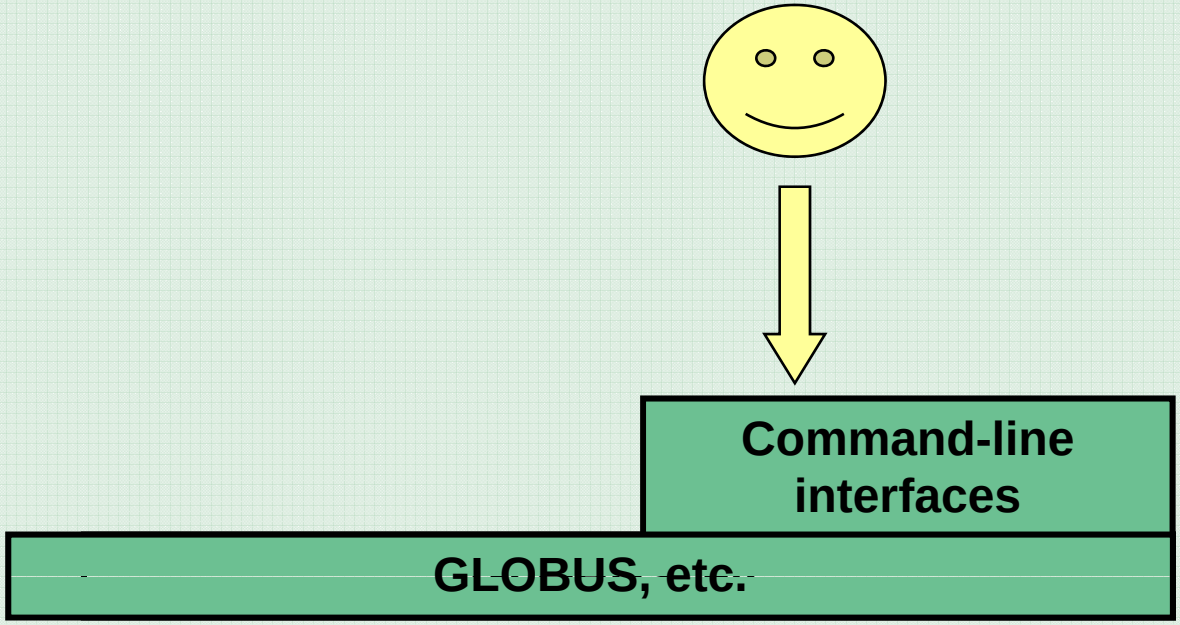
# Overview

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

# CLI Job submission

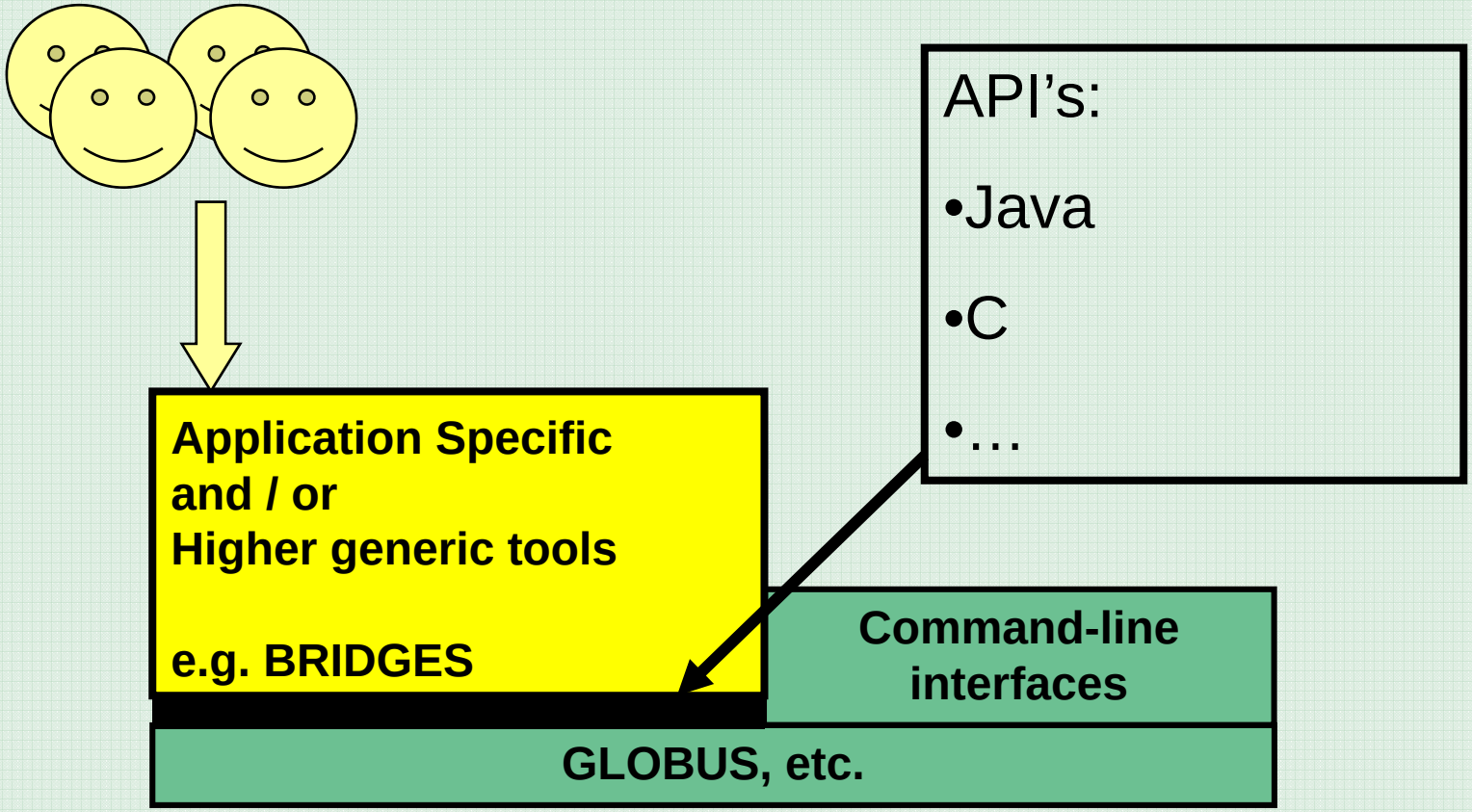
globus-job-run

globus-job-submit/status/get-output




**User's Interface to the grid**

# Application-specific tools




**User's Interface to the grid**

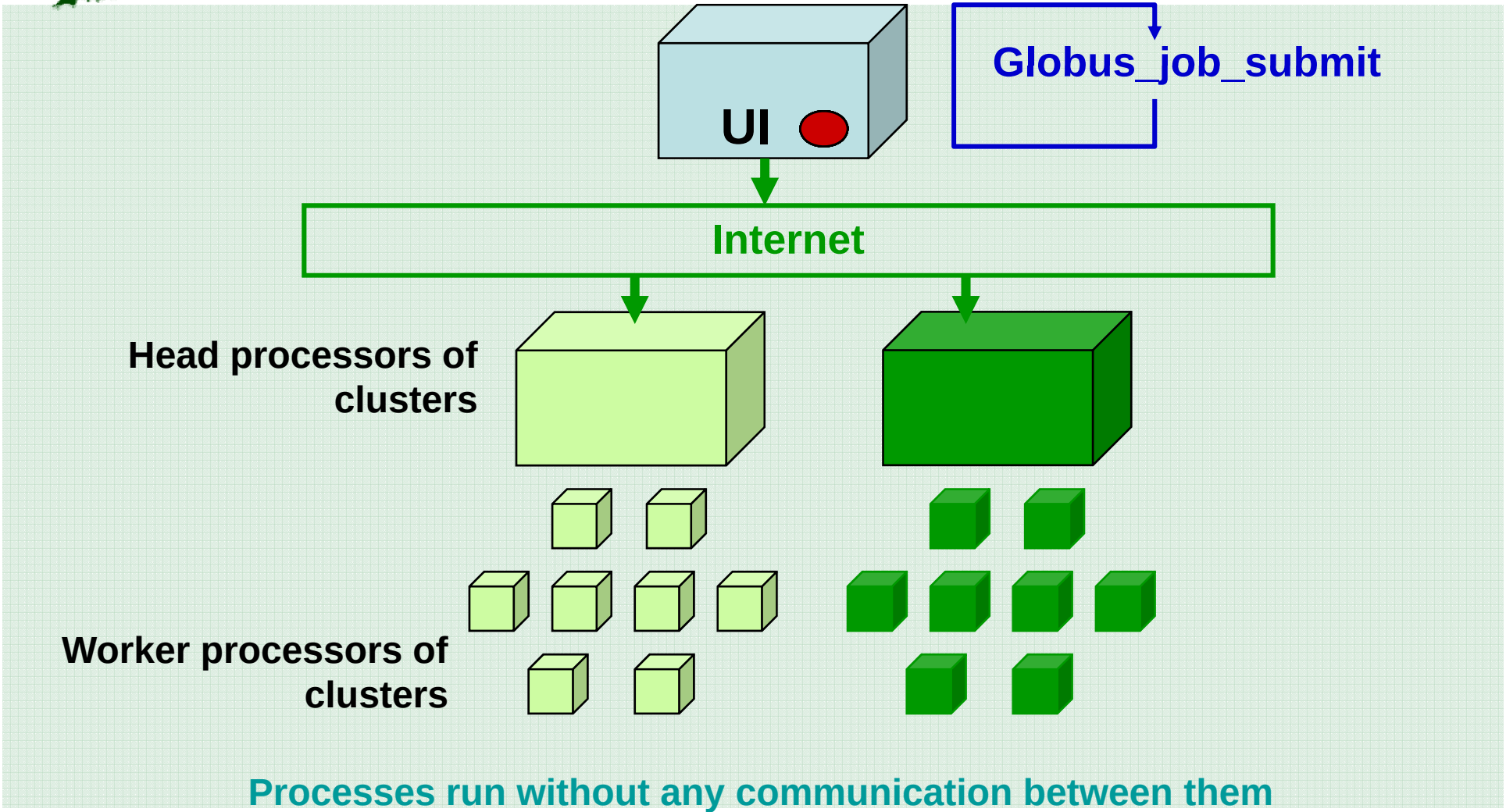
NGS
National Grid Service

# Available API's

- C http://www.globus.org/developer/api-reference.html

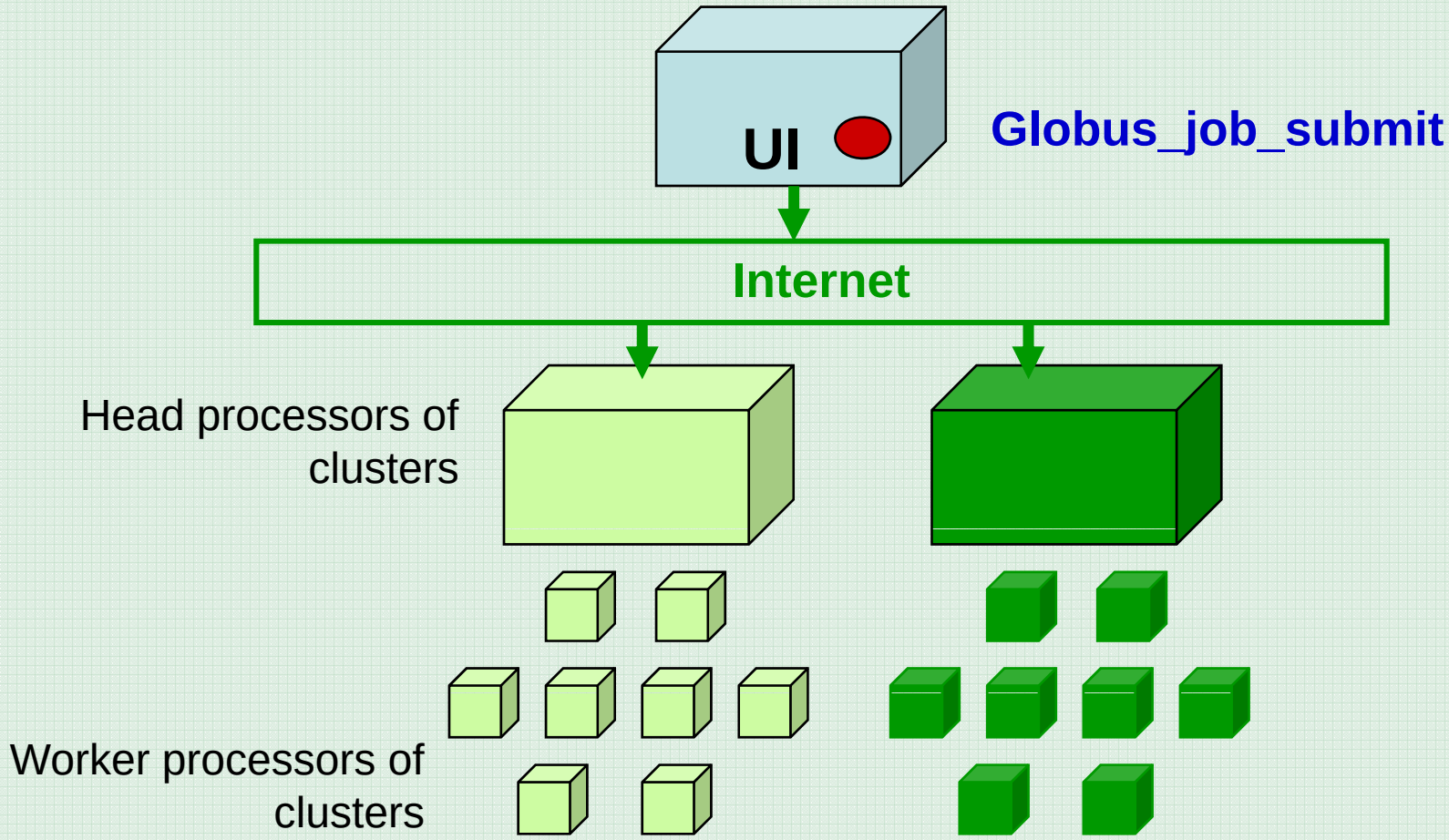
- "Commodity Grid" CoG http://www.cogkit.org/
  - Java, Python, Matlab

# Non-communicating Processes



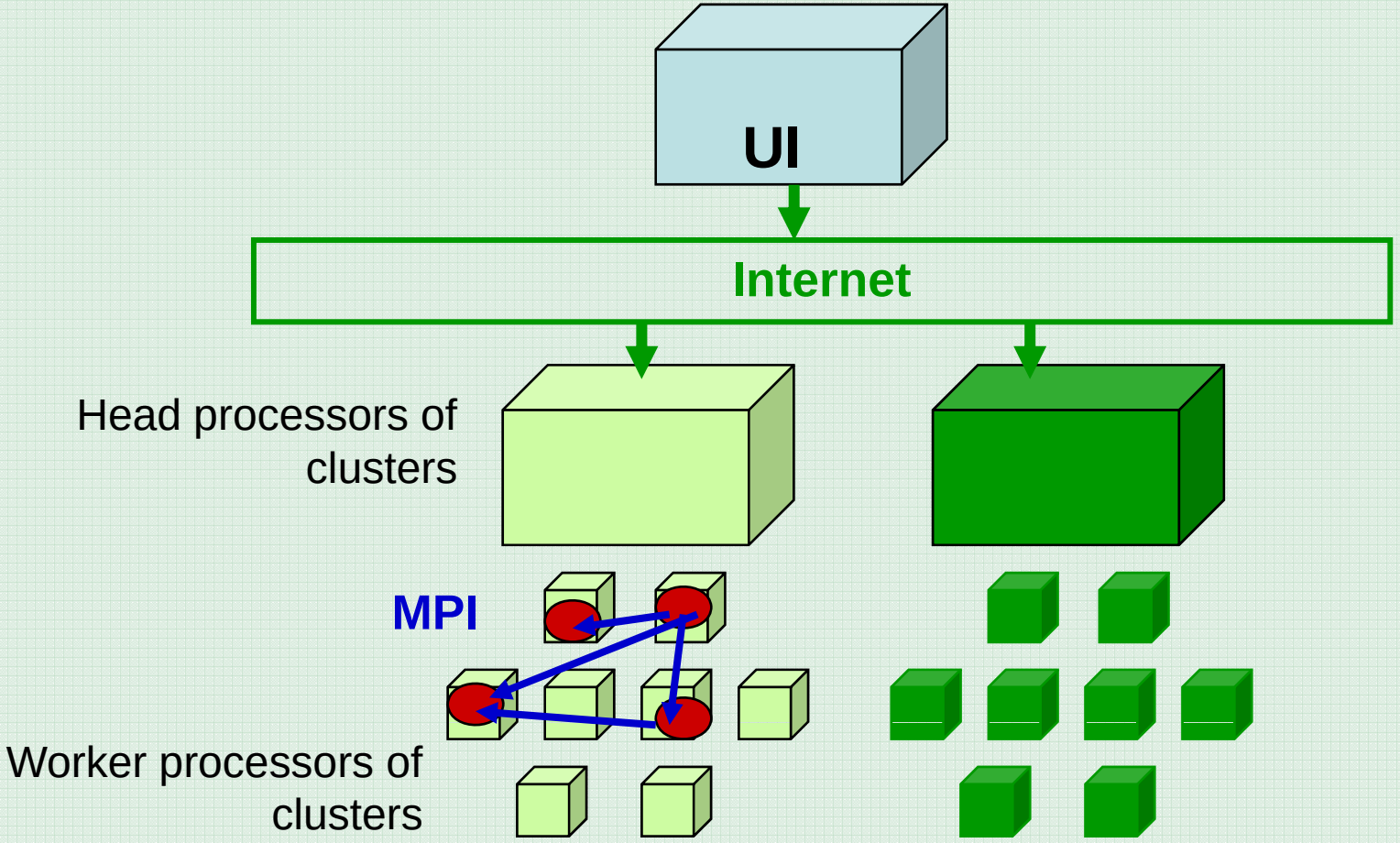

Processes run without any communication between them

# 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**

# Modes of Parallelism

**The NGS nodes open these routes to you – but you have to do a bit of work! (Grid is not magic!...)**

- Non-communicating processes: on NGS, multiple executables run from a script on the UI
- Communicating processes: on NGS, you run one globus-job-submit command – but need to code and build program so it is parallelised
  - MPI for distributed memory
  - OpenMP, multithreading – only on a Cardiff node

NGS National Grid Service

# MPI notes

- 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!

NGS
National Grid Service

# Practical

1. C API Example
2. Java API usage
3. Concurrent processing – from Java
4. MPI example