Preparing the LHC Computing Grid for MPI Applications

Richard de Jong & Matthijs Koot

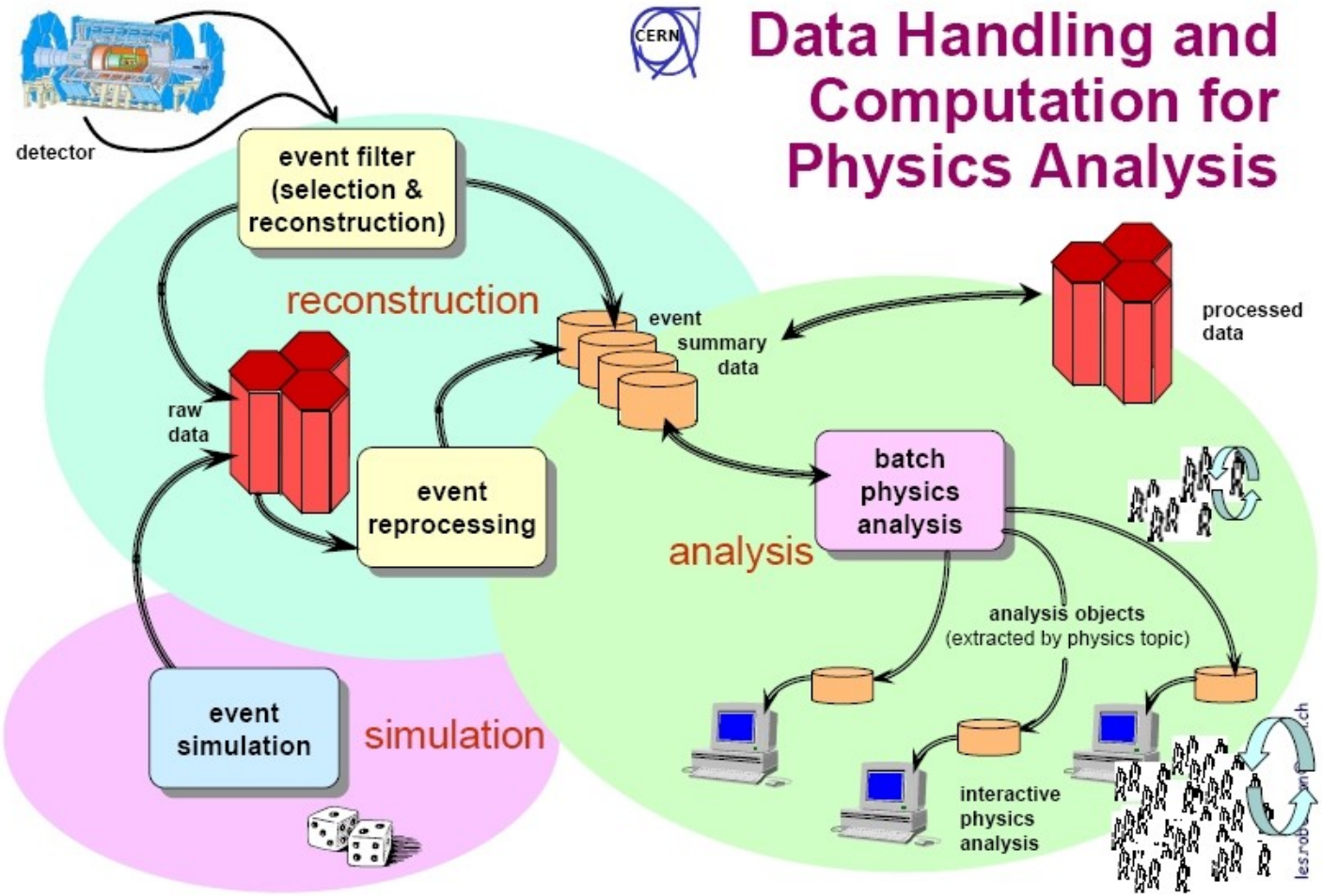
Outline

- **Context**
  - CERN
  - Grid
  - LCG
  - YAIM
- **Problems (and solutions)**
  - YAIM
- **Context (ct'd)**
  - Parallel programming
- **Problems (and solutions) (ct'd)**
  - Single-site MPI
  - Cross-site MPI
- **Conclusion**

Context: CERN & The LHC Project

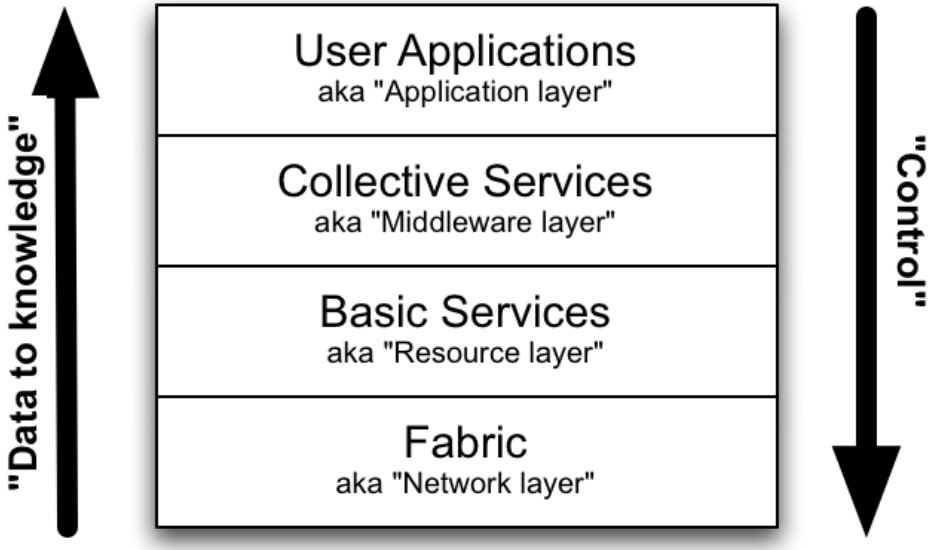
- A new particle accelerator is being built at CERN
- 10 PB/year
- Save (backup) & Analyze all data
- By physicists around the world
- O(8K) nodes at CERN
- But that's not enough…
- Solution: Tier 0, 1, 2

Context: Data flow



Context: Grid

- **A grid is a system that (Foster):**
  - coordinates resources that are not subject to centralized control
  - using standard, open, general-purpose protocols and interfaces
  - to deliver nontrivial qualities of service.
- **Keyword: Middleware**



Context: LHC Computing Grid

- **LCG middleware:**
  - LCG
  - gLite
- **Both based on existing technology:**
  - Globus Toolkit
  - VDT
  - GridFTP
  - Condor
- **Components in the LHC architecture:**
  - User Interface (UI) – To submit a job, retrieve output
  - Resource Broker (RB) – To find a suitable CE
  - Compute Element (CE) – To schedule the job – LRMS
  - Worker Node (WN) – To execute the job
  - Many more…

Context: YAIM

- **YAIM = Yet Another Installation Mechanism**
- **A tool for Grid middleware deployment**
  - Scope
  - Structure
  - Evaluation

YAIM: Scope

- **EDG - WP4:**
  - "provide means to install OS and applications over the network according to policies, to bring a machine in a desired state"
- **Configuration by LCFG(ng), but inflexible and error-prone**
- **So: QUATTOR**
  - Scope on OS and applications
  - But: not (so much) on Grid middleware

YAIM: Scope

- **CERN IT-GD: "should not be so hard" → YAIM**
  - Configuration template
  - For small and/or simple sites
- **But for larger sites?**

YAIM: Structure

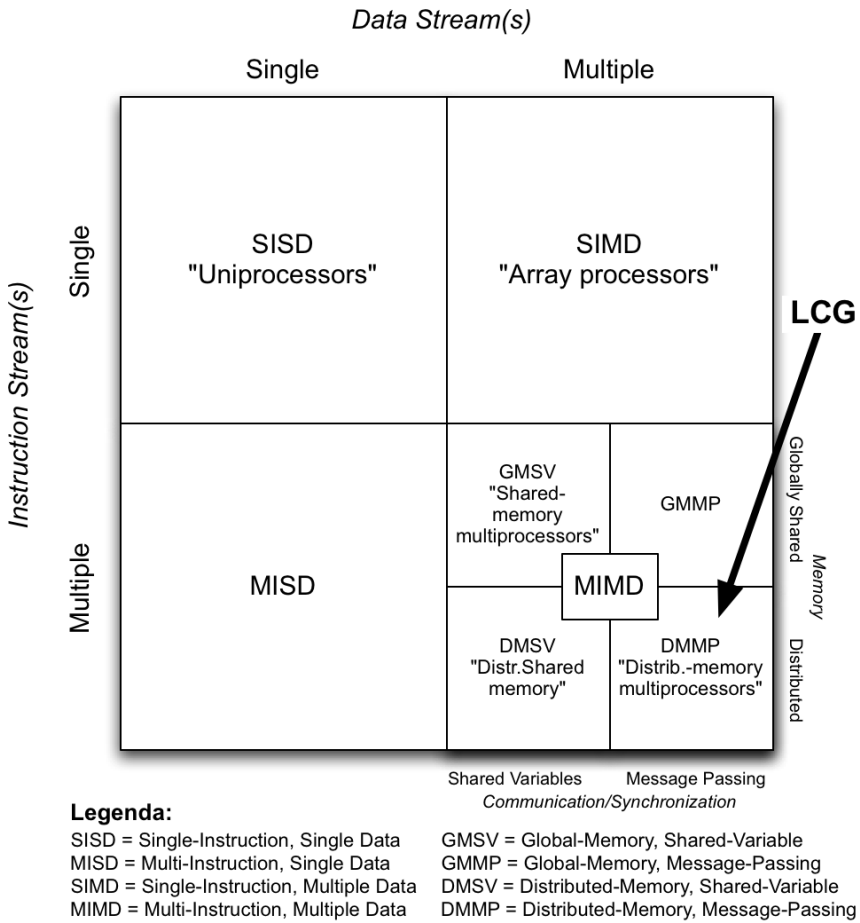
- **Configuration files**
  - site-info.def - Site wide configuration
  - node-info.def - What to configure for which role
- **Scripts**
  - install_node - Install packages
  - configure_node - Configure services
- **Functions**
  - One function covers an atomic piece of configuration
  - One file per function, one function per file
- **Utilities**
  - Helper routines

YAIM: Evaluation

- **Pro:**
  - Straightforward
  - Modular
  - Simple
  - Easy to use
- **Con:**
  - Not atomic
  - Unstructured output
  - Reconfiguring
  - Unconfiguring
  - Users learn to ignore the errors
- **Conclusion: fine for small and/or simple sites**
  - Meanwhile: QWG to integrate YAIM functionality for large sites

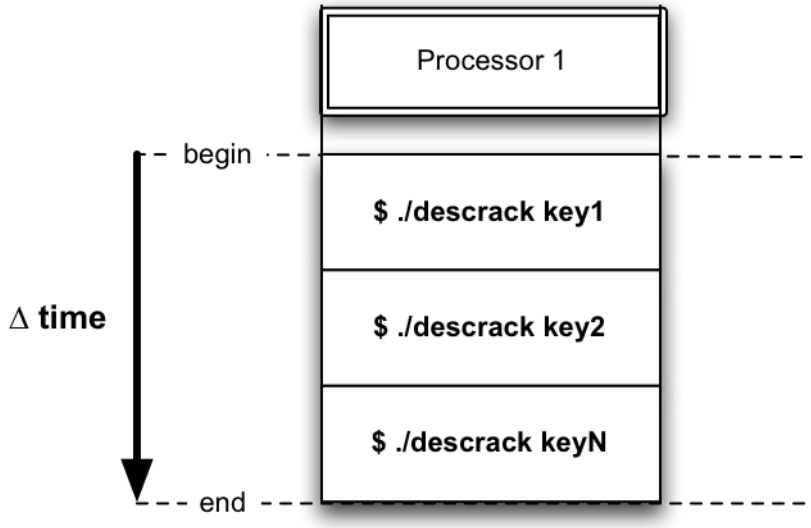
Context: Parallel Programming

- **Traditional:**
  - Use a single processor to solve multiple problems = SISD
- **Parallel programming:**
  - Using multiple processors to solve a single problem
- **Divide the work:**
  - Functional decomposition
    - Each processor a different role = MISD
  - Domain decomposition
    - Each processor different data = SIMD
  - Both
    - Each processor it's own program and data = MIMD

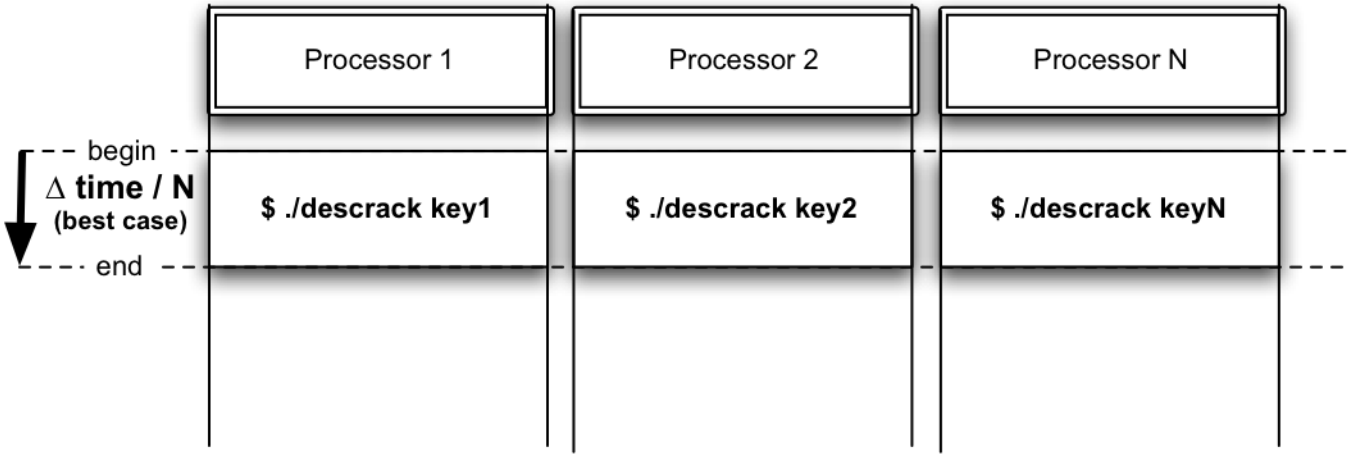


Context: Parallel Programming

- **SISD**

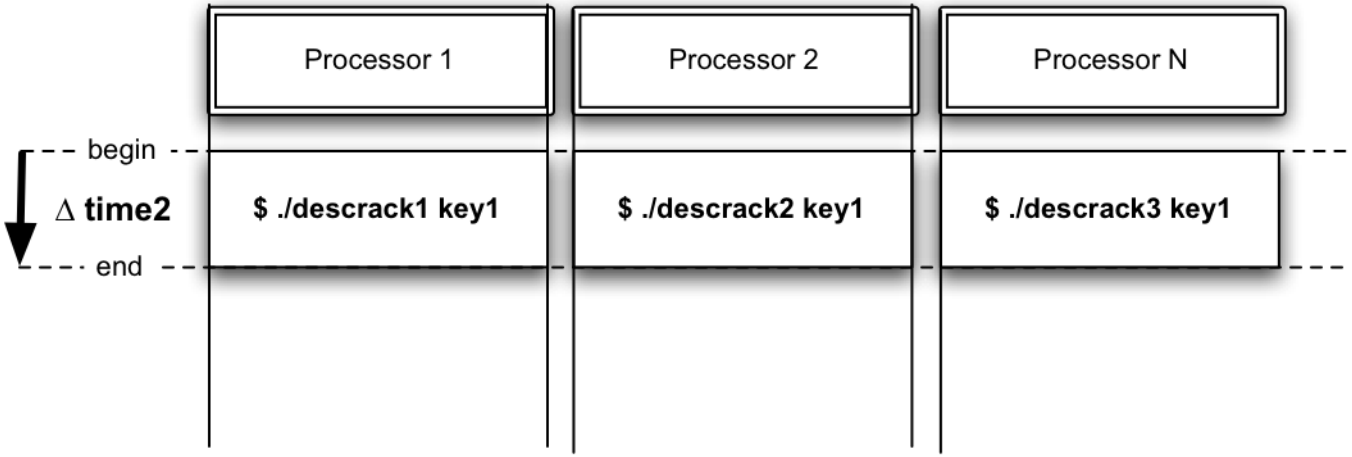


- **SIMD**

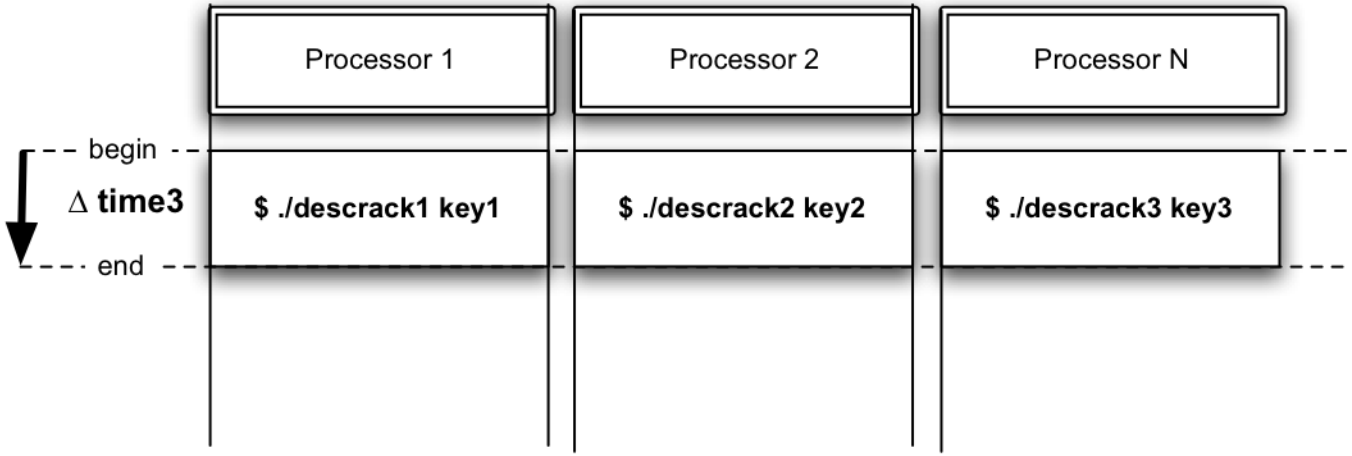


Context: Parallel Programming

- **MISD**



- **MIMD**



Context: MPI

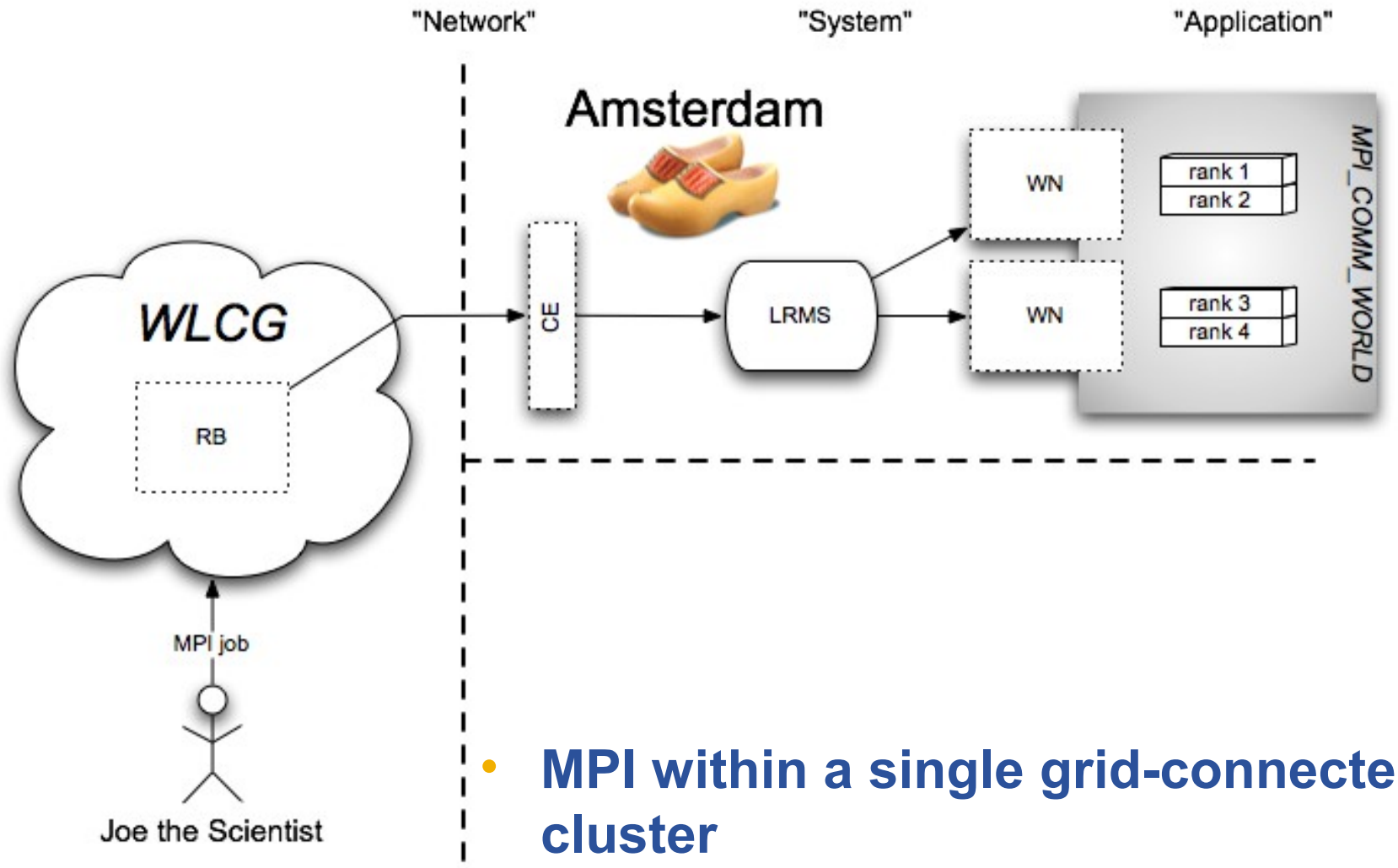
- **MPI = Message Passing Interface**
- **The MPI library is a widely known and used standard for parallel programming**
- **Multiple implementations available**
  - MPICH
  - LAM/MPI
  - OpenMPI
  - MS MPI
  - etc
- **Traffic profile depends on parallel algorithm**

- **Next in line: mrkoot@os3.nl**

Outline

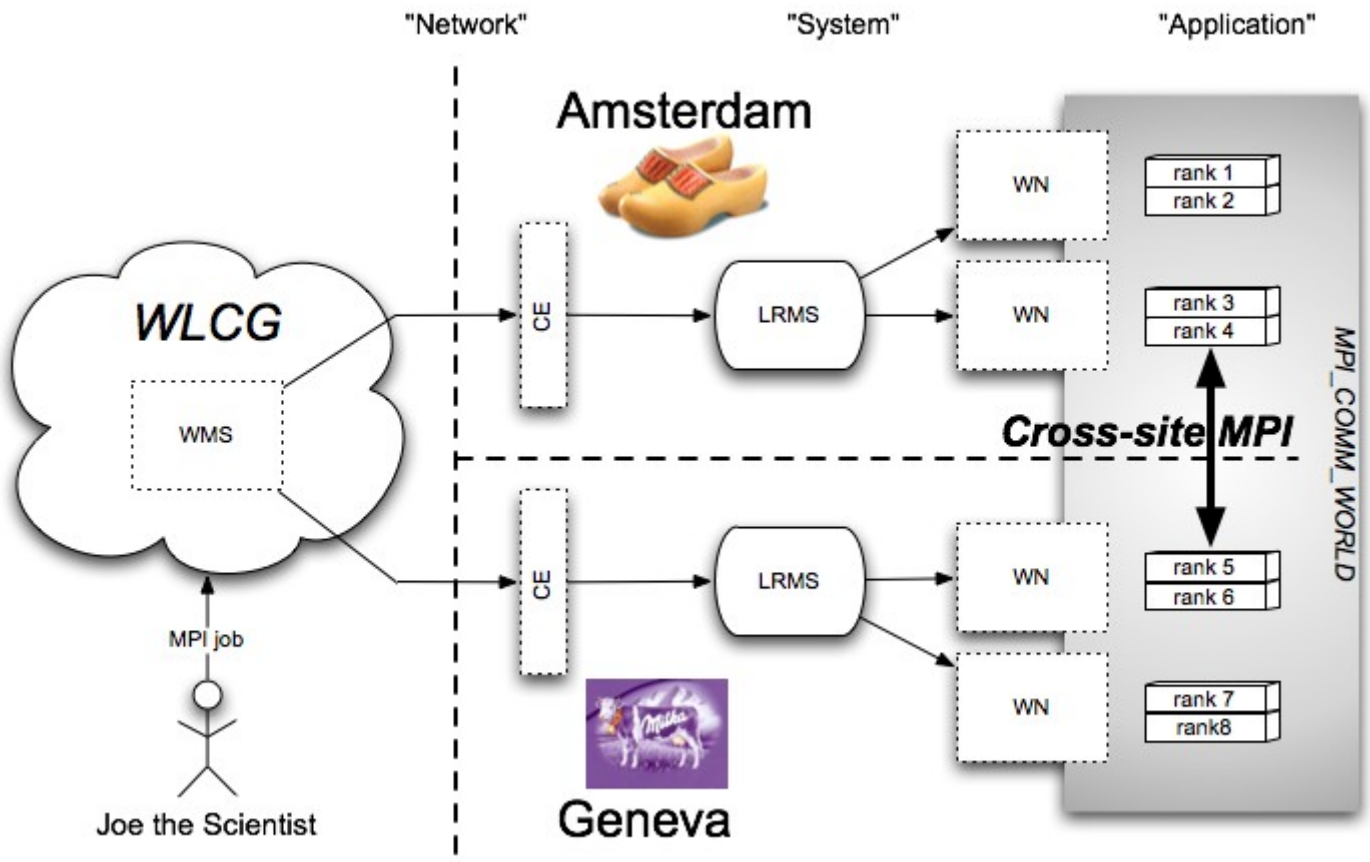
- Context
  - CERN
  - Grid
  - LCG
  - YAIM
- Problems (and solutions)
  - YAIM
- Context (ct'd)
  - Parallel programming
- **Problems (and solutions) (ct'd)**
  - Single-site MPI
  - Cross-site MPI
- **Conclusion**

Problems: Single-Site MPI

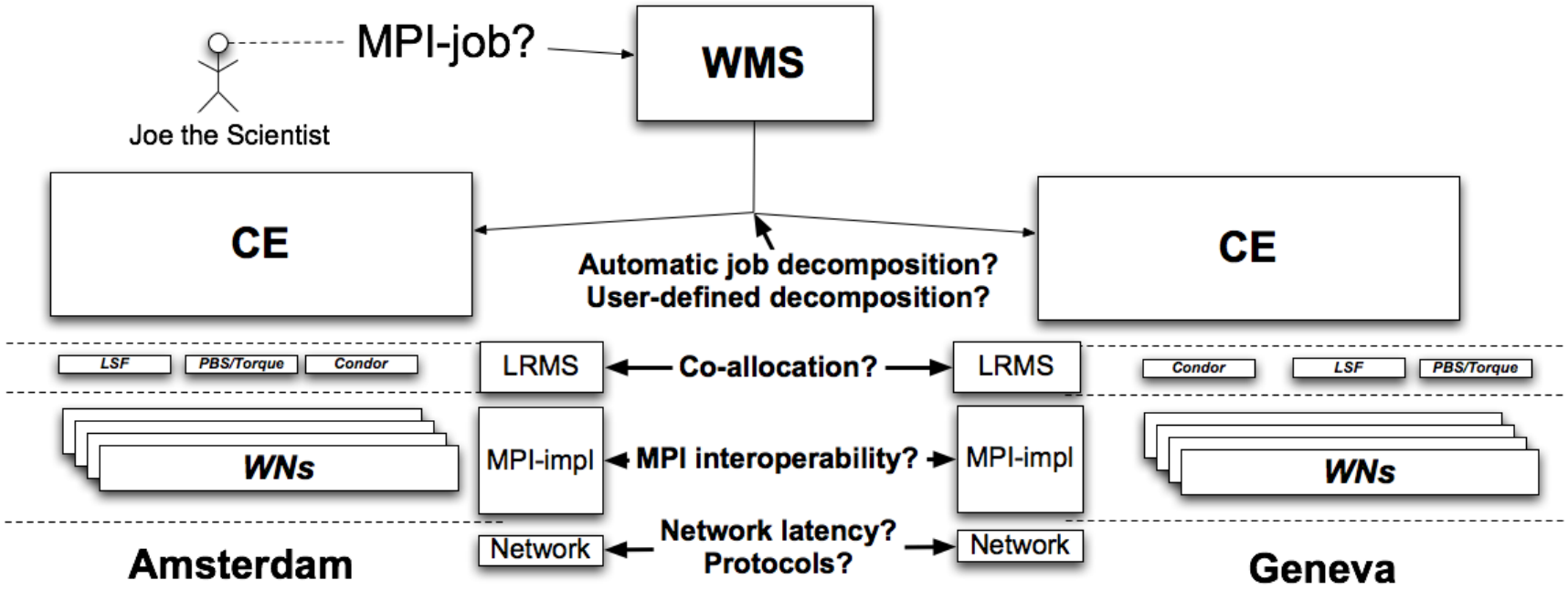


- MPI within a single grid-connected cluster
- Integration between MPI and LRMS

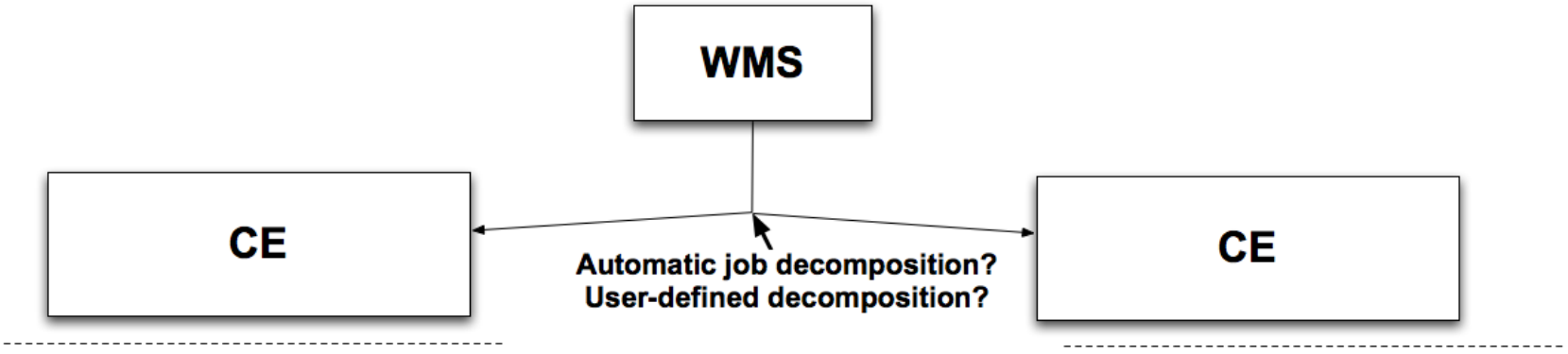
Problems: Cross-Site MPI



Problems: Cross-Site MPI

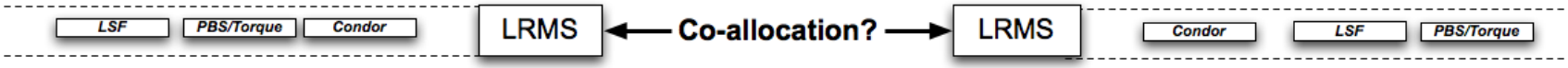


Problems: Cross-Site MPI



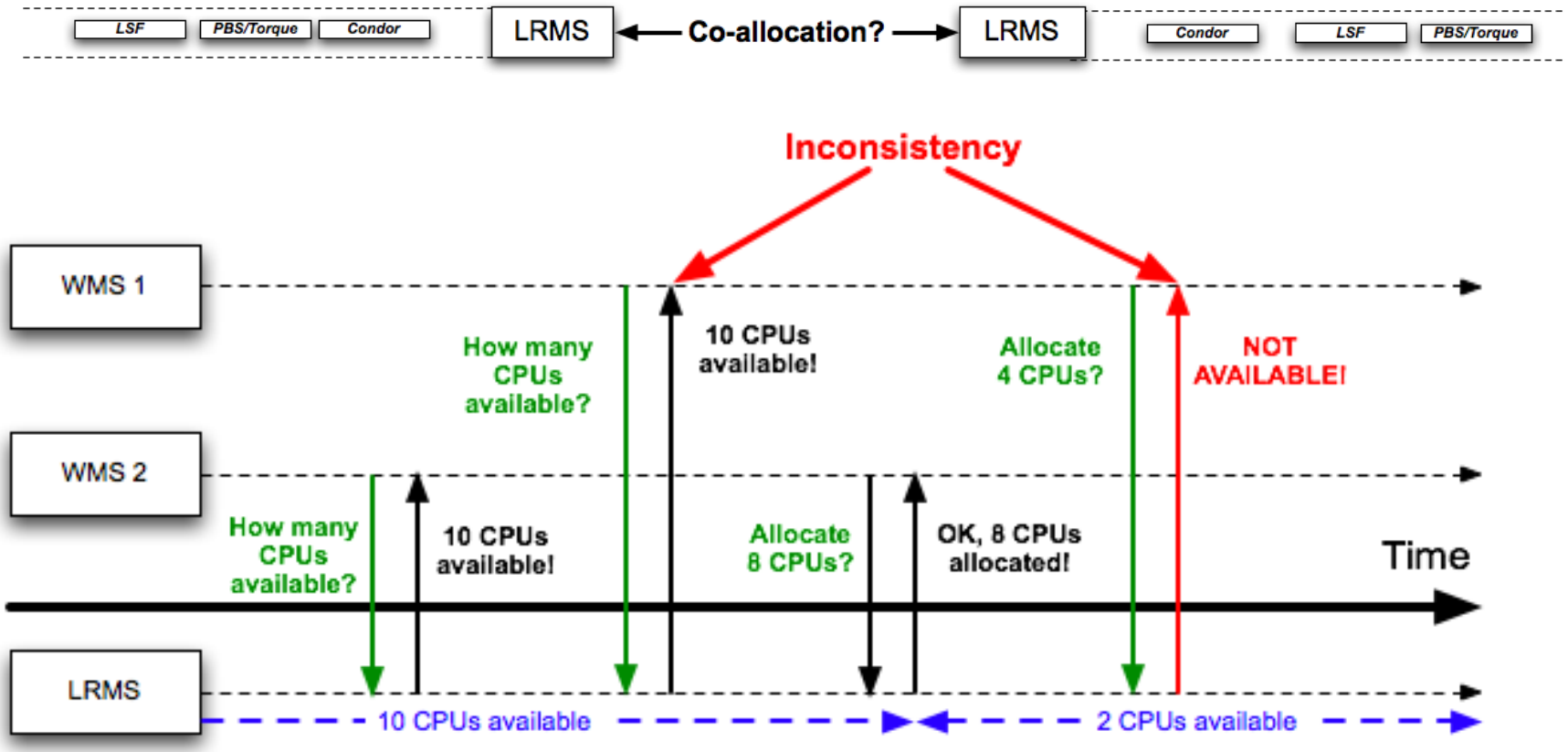
- **Job decomposition**
  - Functional ~, domain ~
  - Is it possible to automate?

Problems: Cross-Site MPI

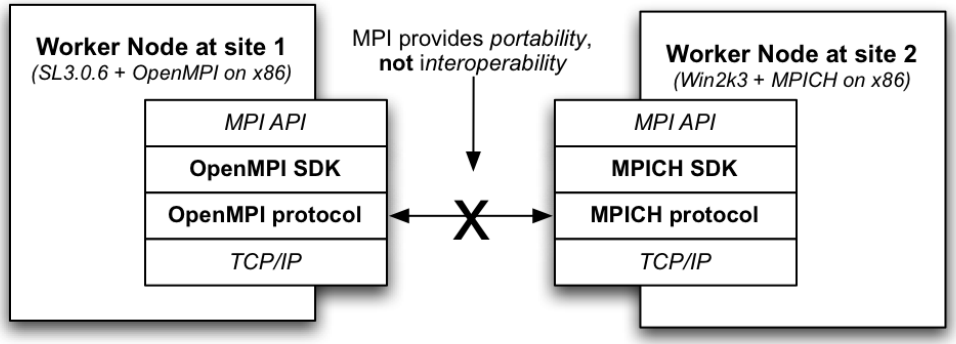


- **Co-allocation**
  - Resource scheduling is tough!
  - Learn from Koala in DAS-2?
  - LRMS interoperability?

Problems: Cross-Site MPI



Problems: Cross-Site MPI

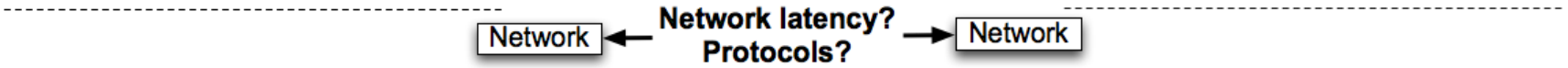


- **MPI interoperability**
  - MPI is *portable*, not *interoperable*
  - Sites should provide common MPI

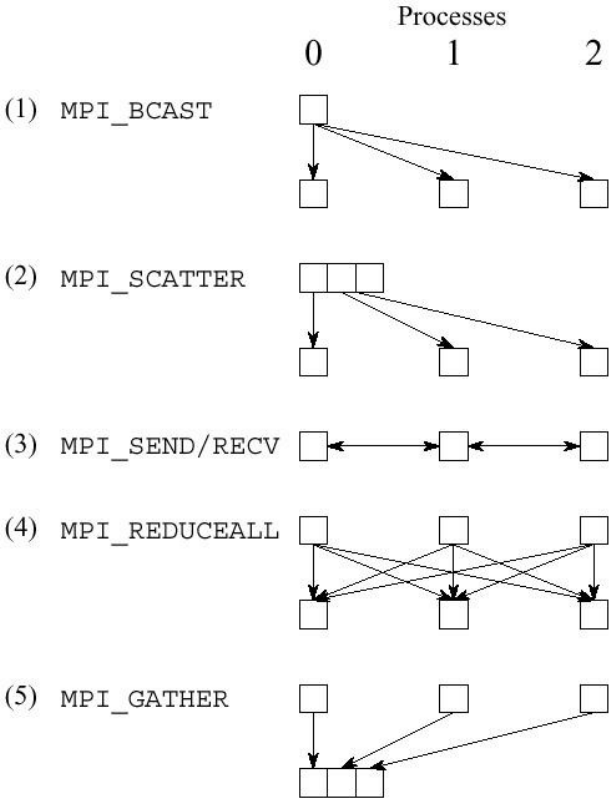
Problems: Cross-Site MPI

- **Networking**
  - Latency(-tolerance?)
  - Protocols (public IP addresses?)
  - Security (dynamic firewall with VO-EAP?)

Problems: Cross-Site MPI



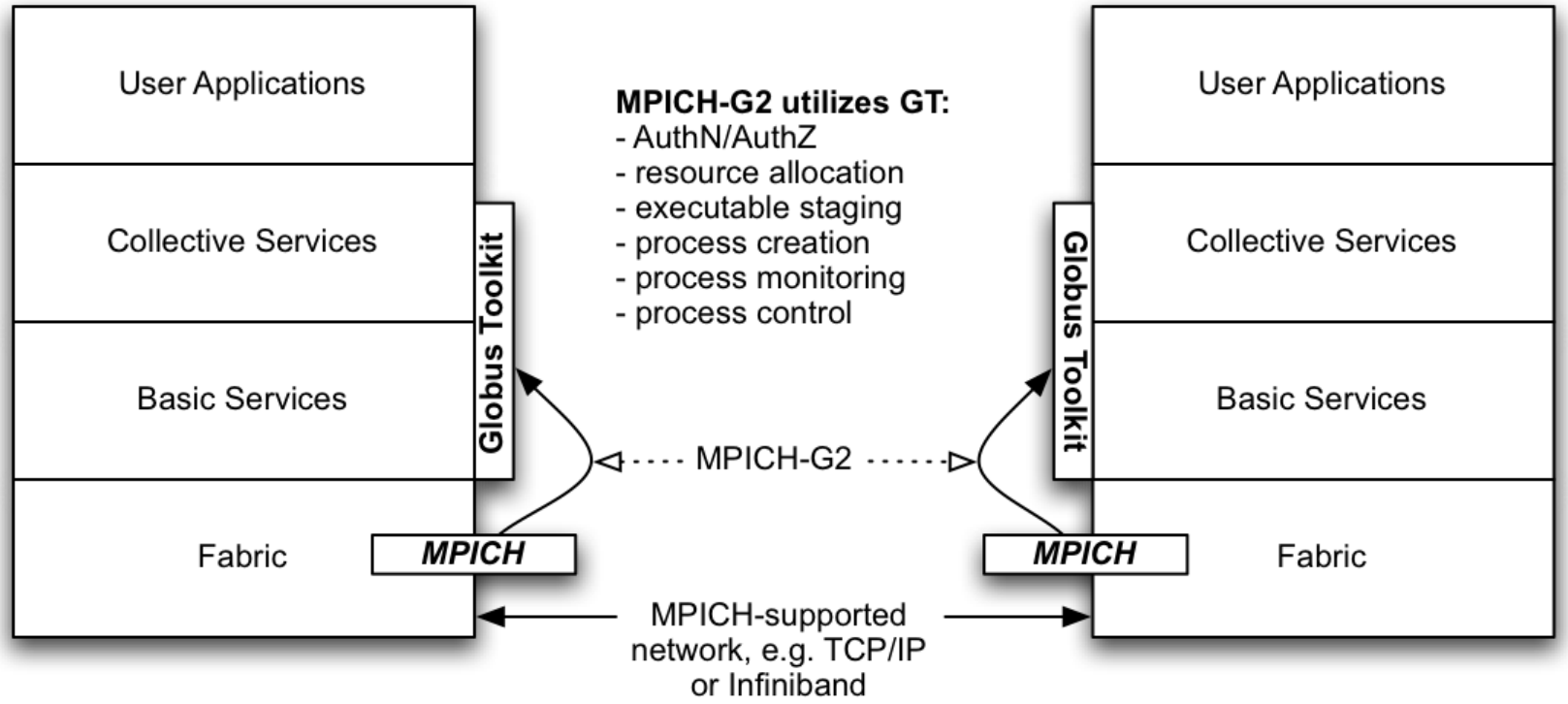
- **Networking (2)**
  - Collective ops?



Problems: Cross-Site MPI

- Integration is needed between MPI and grid middleware (AuthN/Z, monitoring)
- MPICH-G2 demonstrates integration between MPICH and Globus
- Might (still?) be usable on LCG

Problems: Cross-Site MPI



Problems: Cross-Site MPI

- **It *has* been demonstrated**
  - Legion (2001)
  - TeraGrid (2003)
  - K*Grid in Korea (2005)
- **…but not yet over a high-latency, heterogeneous, Internet-based grid**

Conclusion(s)

- **Single-Site MPI is probably OK**
- **Cross-Site MPI is tough, but possible**
  - Good software engineering required
- **YAIM needs improvement**

Questions

- **Questions?**