

M-grid: Linux in the First Production Grid Environment in Finland

Arto Teräs <arto.teras@csc.fi>

Linux & Open Source training

Hotel Kämp, Helsinki, November 1, 2005

(English version of the slides presented in Finnish)



Contents

- **CSC and the Finnish Material Sciences Grid (M-grid)**
- **Standard package or a custom solution?**
- **Managing installation and updates**
- **Shared system administration — can it work?**
- **User experiences**
- **Grid use and resource sharing**
- **Security challenges**
- **Conclusions**



CSC — the Finnish IT center for science

- **Mission: National-level IT services for research and education, development and maintenance of the IT infrastructure**
- **Fields of service:**
 - Funet services
 - Computational services
 - Applications: software and databases
 - Information systems management
 - Expertise in scientific computing
- **Customers: Universities and polytechnics, research institutes and their staff who use information technology**
- **Owned by: Ministry of Education**

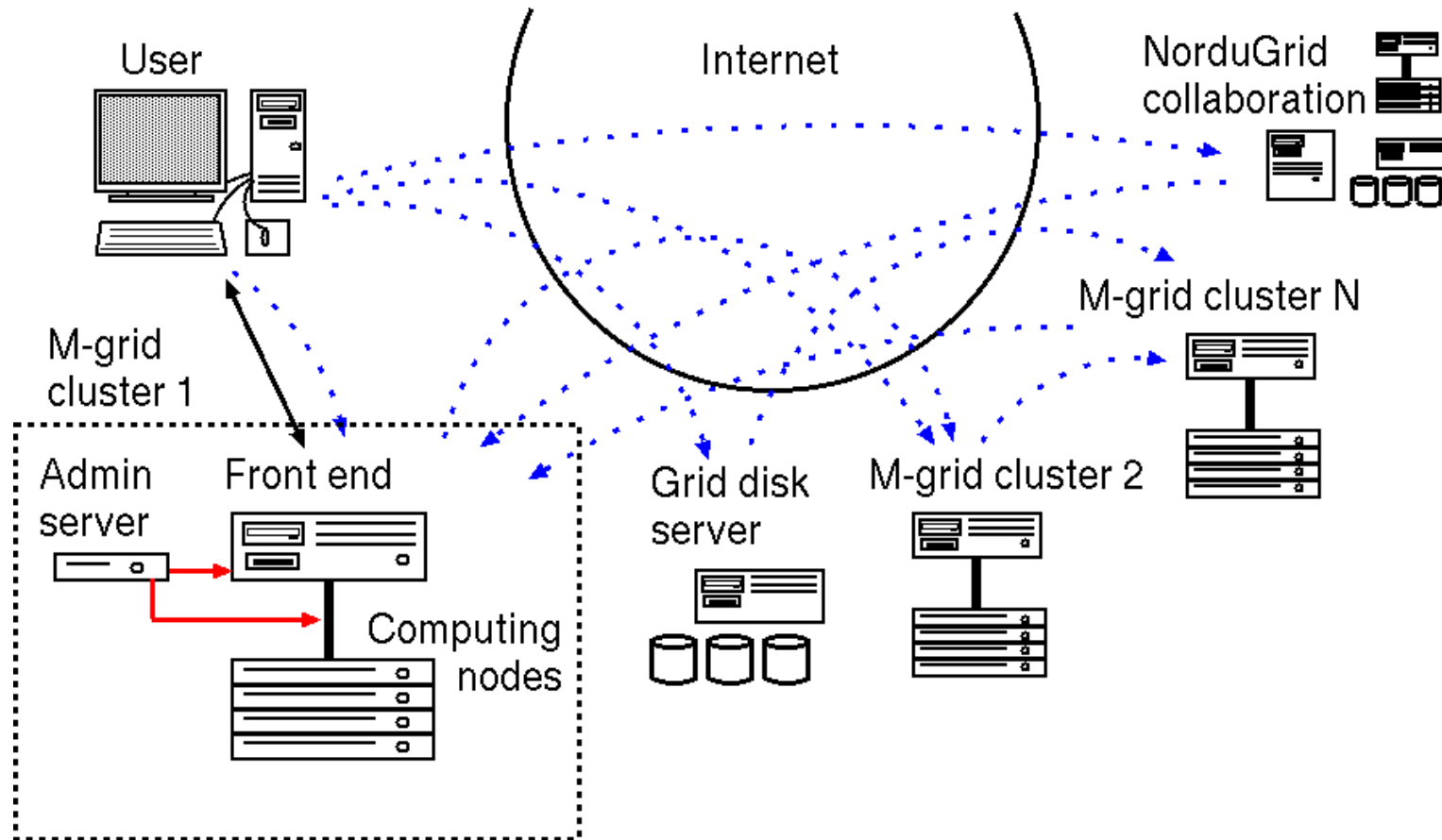


Materiaalitutkimuksen grid (M-grid)

- **Goal: Throughput computing capacity mainly for the needs of physics and chemistry researchers**
- **Joint project between seven Finnish universities, Helsinki Institute of Physics and CSC**
 - Partners mainly laboratories and departments, not university IT centers
- **Jointly funded by the Academy of Finland and the participating universities**
 - Funding application Nov 2003, deployment Oct 2004
- **First large initiative to put Grid middleware into production use in Finland**
- **Platform: Linux based PC clusters**

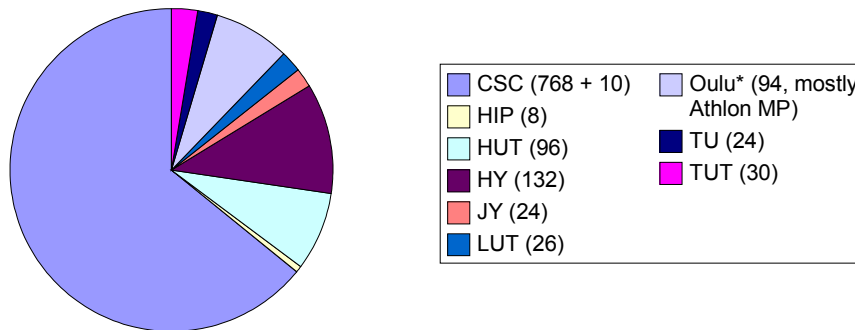


Grid environment



Hardware and CPU distribution

- **Ten clusters of varying size**
 - Dual AMD Opteron computing nodes (HP DL145): 1.8-2.2 GHz, 2-8 GB RAM, 80-320 GB local disk
 - Front end (HP DL585): 1-2 TB shared disk
 - Network 2 x Gbit Ethernet + remote administration network
- **Total 778 (CSC) + 434 (universities) CPUs in the computing nodes, theoretical total computing power 5 TFlop/s.**



Operating system and Grid middleware



- **NPACI Rocks Cluster Distribution**

- Cluster oriented Linux distribution, main developer San Diego Supercomputing Center, U.S.A.
- Based on Red Hat Enterprise Linux, but not a Red Hat product
- <http://www.rocksclusters.org>



- **N1 Grid Engine batch queue system**

- Local resource management in each cluster

- **NorduGrid ARC Grid middleware**

- Enables shared use of the systems, the middleware selects a free resource automatically
- <http://www.nordugrid.org>



Standard package or a custom solution?

- **Linux was an easy choice — already the leading OS in computing clusters**
- **Both commercial and noncommercial options available for cluster management**
 - Our choice was Rocks: no commercial support but a relatively large user base and dedicated development team
 - Solutions offered by system vendors perhaps better integrated, but independence and ability to customize also important
- **A complete turn-key solution didn't exist**
 - Open source product gave the possibility to study and add own modifications in advance independently of the hw vendor choice
- **Reliability requirement: stable base environment and local use, more experimental grid environment**

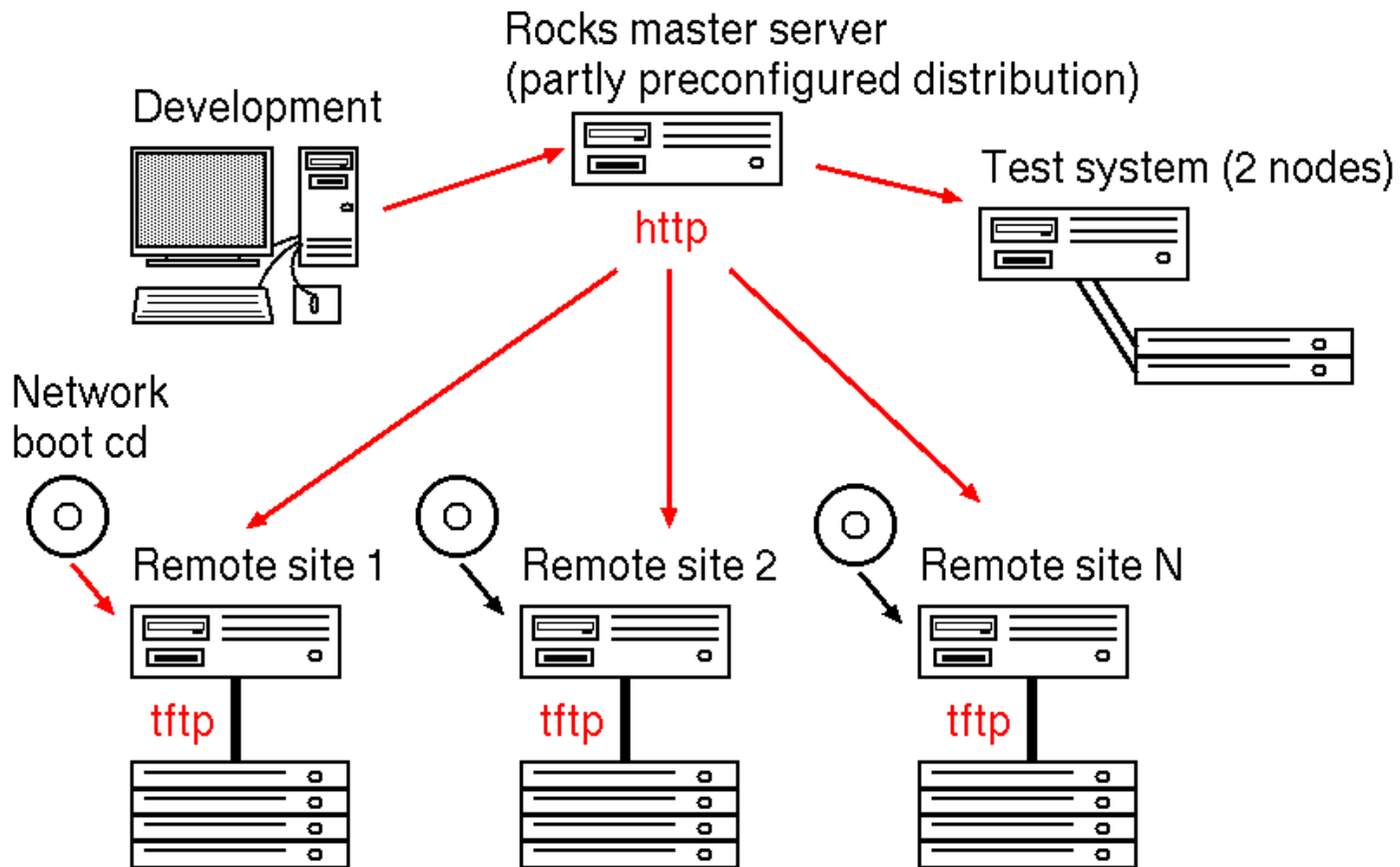


System administration in M-grid

- **Tasks divided between CSC and site administrators**
- **CSC administrators:**
 - Maintain (remotely) the operating system, batch queue system, Grid middleware and certain libraries for all sites except Oulu
 - Separate small test cluster for testing new software releases
- **Site administrators**
 - Local applications and libraries, system monitoring, user support
- **Regular meetings of administrators every two months, common mailing list**



Installation

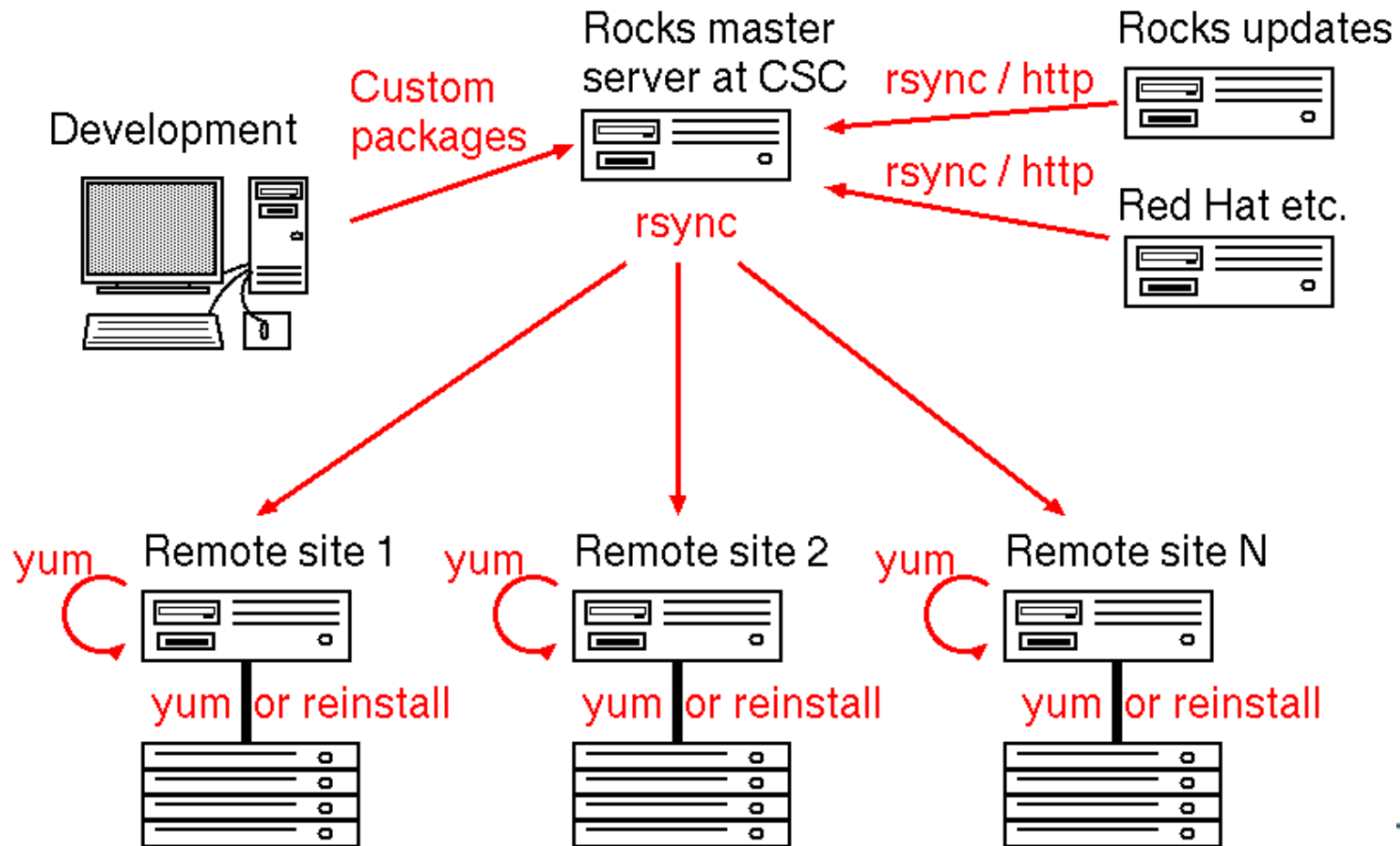


Deployment experiences

- **Hardware installation by the technicians of the vendor**
- **CSC prepared the distribution and a boot cd, local administrators responsible for installing their own cluster**
- **Preparing the distribution took more time than expected**
 - Hints for configuration and modifications from the Rocks mailing list as is common in the open source community
- **Actual deployment went rather smoothly**
 - Most sites spent less than a day installing the OS and nodes, larger sites took two days
 - One site had strange problems taking more time
- **A few settings especially concerning MPI parallel runs needed to be fixed manually afterwards**



Installing updates



Rocks pros and cons

Good:

- Easy to get started, designed for clusters
- Nice monitoring tools, many things work out of the box
- Most major vendors have their hardware certified for RHEL
=> Rocks usually works too

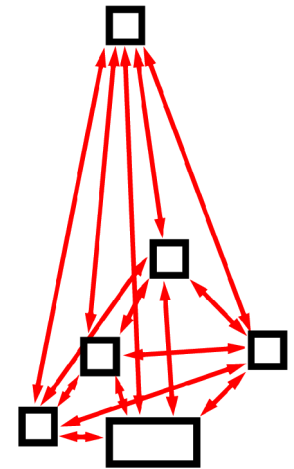
Something to improve:

- The Rocks team does not publish their own security hotfixes and commercial support is not available
 - Red Hat source rpms or binaries from RHEL clones usually work
- Diagnosis and debugging difficult when customizing the distribution



Goals of Shared System Administration

- **Centrally administered foundation while maintaining local control**
 - A new paradigm -- traditionally in Finland academic HPC resources have been centralized at CSC
- **Easier for universities than setting up their own cluster from scratch**
 - However, needs a significant amount of work both from CSC and the local sysadmins
- **Take advantage of the local sysadmin expertise**
 - Site administrators know the software of their own group best => faster and better user support



36 pairs for collaboration!



Positive experiences

- **Site administrators have found CSC support valuable**
 - On the other hand local control (root access) enables quick fixes and is important psychologically
- **Site administrators have picked up tasks which benefit everyone — CSC has not done everything**
- **Collaboration has strengthened relationships between groups also in their research**
- **Systems are close to the user**
 - Easier to talk to the own group sysadmin, less support requests to CSC
- **Most site administrators are also users => direct usability feedback to CSC**



Negative experiences

- **Configuring the Sun Grid Engine v. 5.3 batch queue system**
 - Current version 6.0 is more suitable for clusters
- **Wiki based FAQ hasn't become popular, questions and answers are buried on the mailing list**
 - The Wiki model can also be a success: e.g. Wikipedia
- **Gaps in the user documentation**
 - Mainly due to lack of human resources
 - Documentation can be written in a distributed group but compiling it needs central coordination
- **Some users found support poor**
 - Varying experiences: on some sites users are very happy



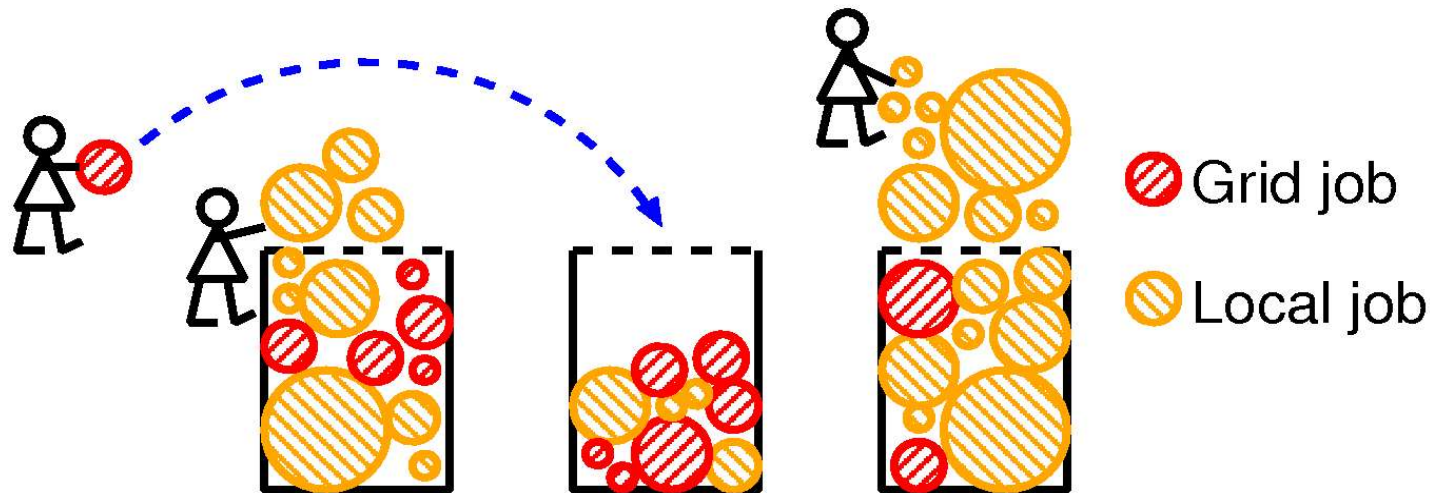
User experiences during the first year

- **Users got started relatively quickly: after a few months the average load was over 50%, currently close to 100%**
 - Linux was already a familiar environment for most users
- **Performance has been quite satisfactory**
- **Reliability has been mainly good**
 - Front ends had stability problems in the beginning, MPI runs are sensitive to changes in the environment
- **Choosing the Fortran compiler was difficult**
 - GNU Fortran compiler works but produces slow code: Pathscale now the recommended one
 - Some applications compatible only with some specific compiler



Grid use and resource sharing

- **Policy: Jobs can be submitted both to the local queue and through the grid interface**
 - Queue priority: local jobs 80%, grid jobs 20%
- **Goal is to minimize waste of resources: empty nodes are always available for use (dynamical sharing)**

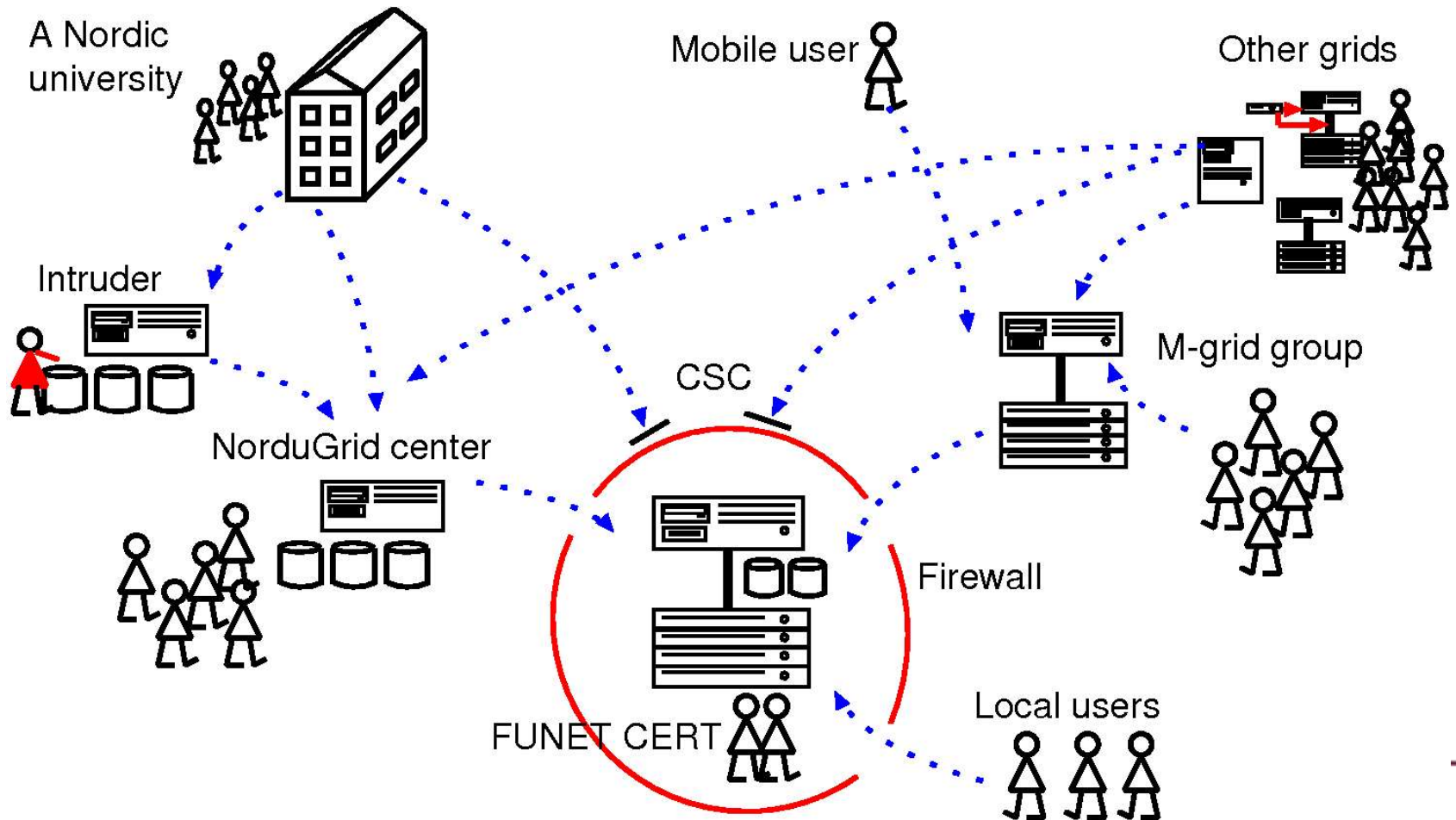


Grid experiences

- **Grid use started August 2005**
 - Installation was delayed due to other tasks and a few technical problems
 - Environment still in development
- **Grid environment must be better than the existing one, otherwise nobody will use it!**
 - Long queue in the local cluster and empty resources on the Grid may be a good enough incentive
- **Currently only a few Grid users, time will show how well the Grid environment will be adopted**
- **Collaboration model has been successful: Grid projects always have other aspects than just the technology**



Grid collaboration and security



CSC

Security challenges in the Grid

- **Grid goes beyond organizational borders**
=> Mutual trust is a key requirement!
- **A few new threats and all the old ones with an extended scope**
 - A single compromised user account still the easiest way to break into the system
 - An user account in grid is a pass to a large number of resources
- **Systems with hundreds of users are always a risk**
 - Compromises cannot be completely prevented in the long term: need to concentrate in detecting them quickly
 - Clear operating procedures for incident response necessary



Security challenges (continued)

- **Getting all the relevant parties involved**
 - Computing centers, university IT departments, local admins, CERTs and also users
 - International collaboration
- **Defining responsibilities important to establish trust**
 - Risk analysis
 - Acceptable use policy and user account administration
 - Incident response
 - Data protection and privacy



Conclusions

- **Sharing system administration tasks can work**
 - Personal contacts are important — face to face meetings are the best way to avoid flame wars
- **User support in a distributed system potentially very good but needs special attention**
- **A complete turn-key solution not available: chose a base which can be extended and built on**
- **Grid projects strenghten ties between groups also independently of the technology**
- **Grid goes beyond organizational borders: not possible without mutual trust**



More information

- M-gridin homepage: <http://www.csc.fi/proj/mgrid/>
- Rocks homepage: <http://www.rocksclusters.org>
- NorduGrid homepage: <http://www.nordugrid.org>
- Contact people:
 - Arto Teräs <arto.teras@csc.fi>
 - Kai Nordlund <kai.nordlund@helsinki.fi>
 - Olli-Pekka Lehto <oplehto@csc.fi> (Rocks)
 - Urpo Kaila <urpo.kaila@csc.fi> (security)
- Thank you! Questions?

