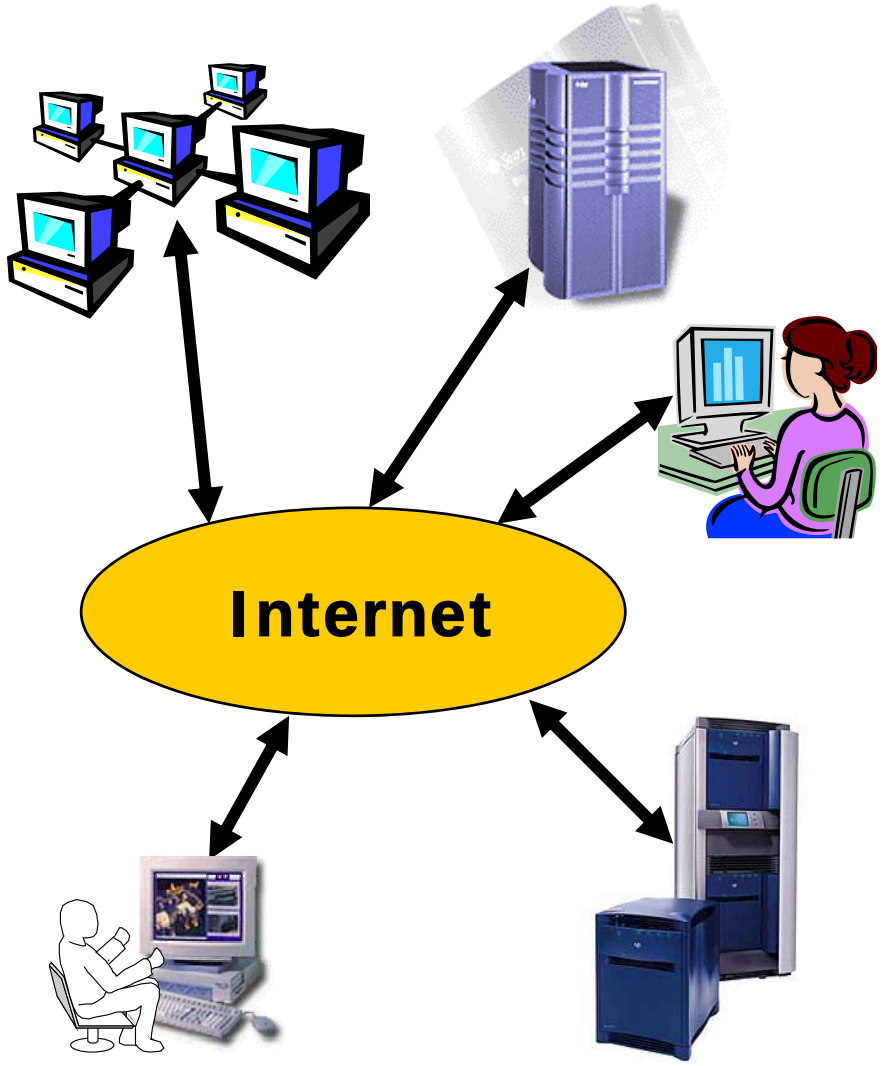
Introduction to Grids and Grid applications

Gergely Sipos

MTA SZTAKI
www.lpds.sztaki.hu
sipos@sztaki.hu

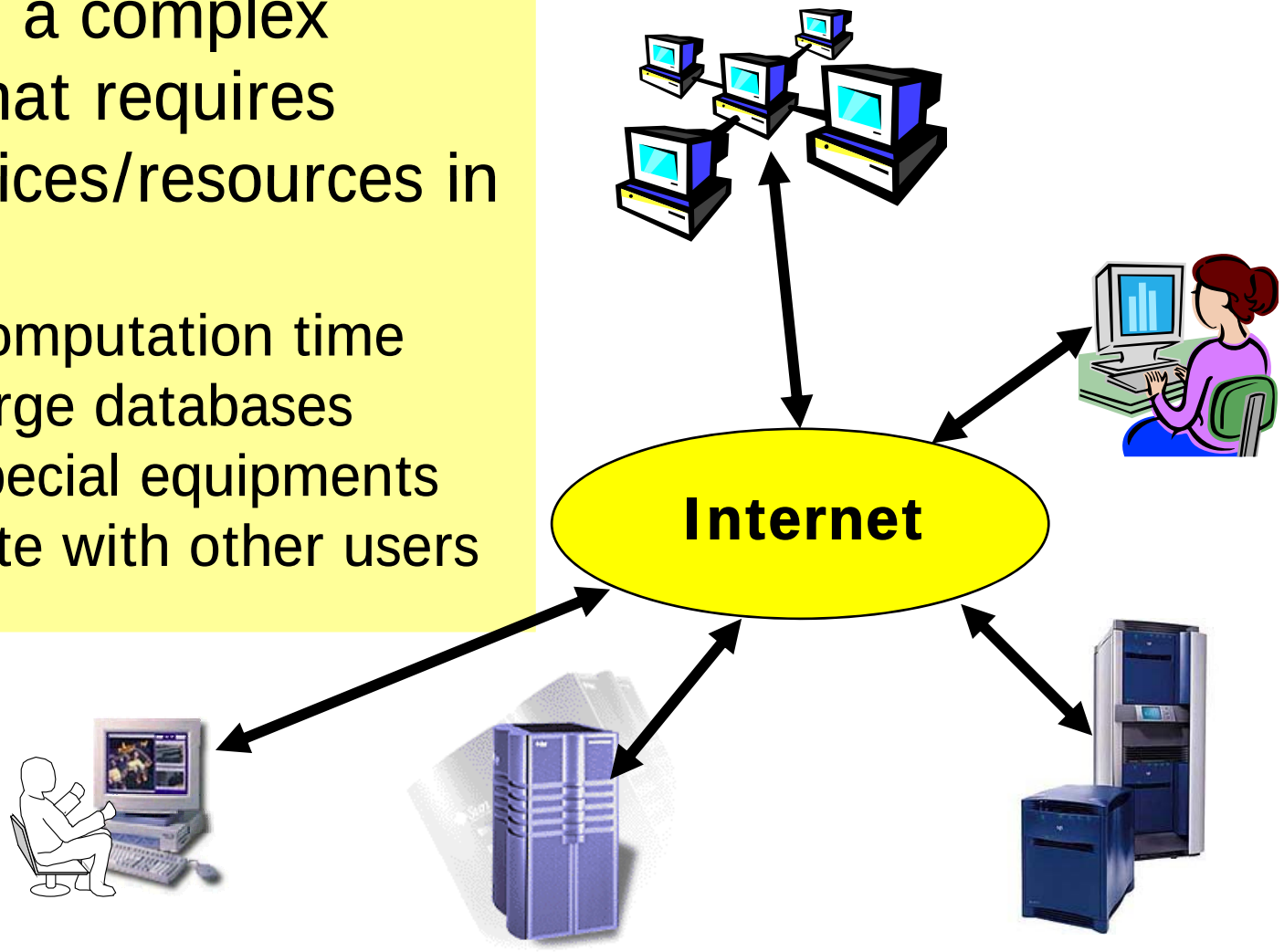
What is Grid?

- A Grid is a collection of computers, storages, special devices, services that can **dynamically join and leave** the Grid
- They are **heterogeneous** in every aspect
- They are geographically **distributed** and connected by a **wide-area network**
- They can be accessed **on-demand** by a set of users



Why use a Grid?

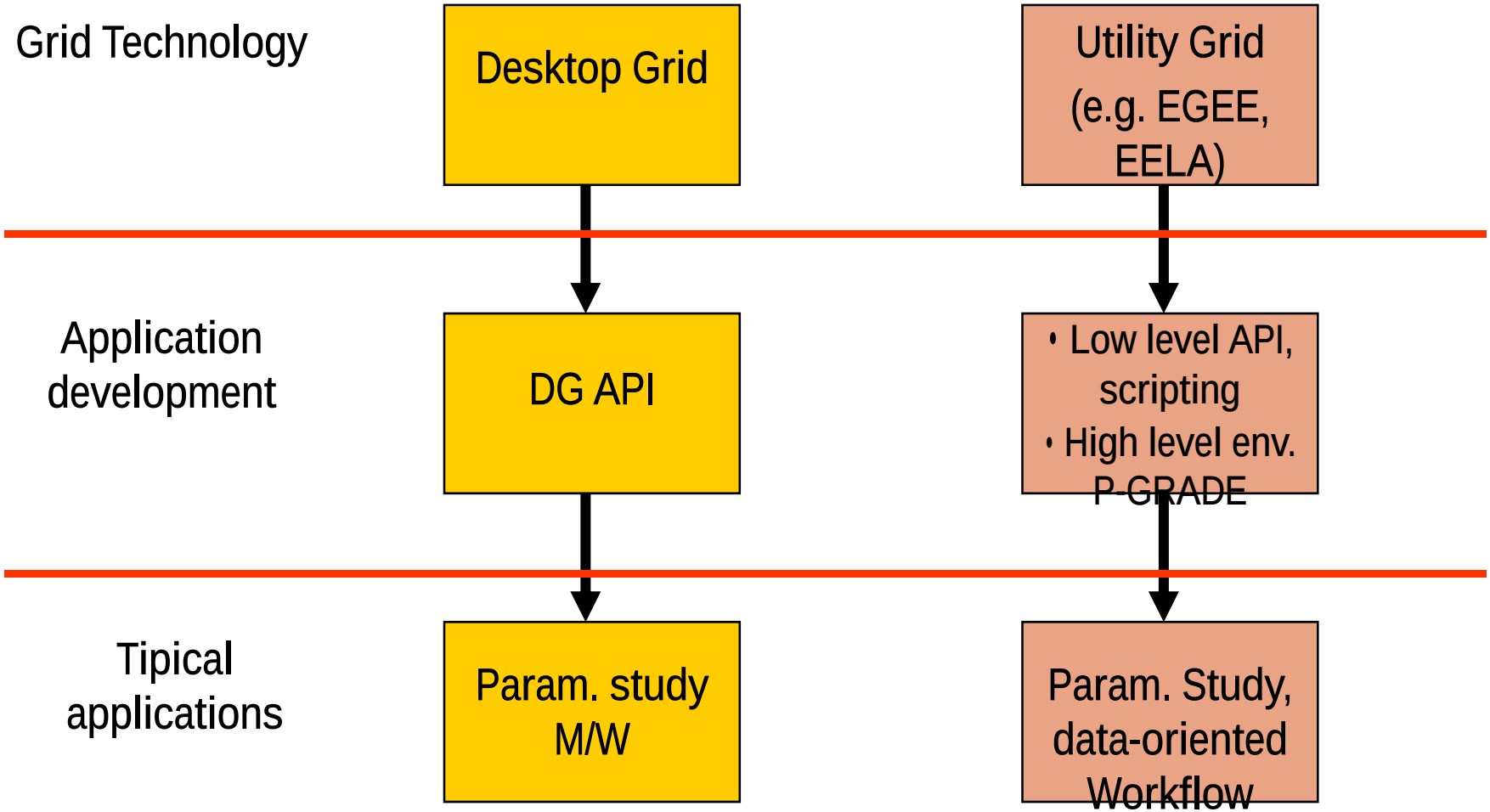
- A user has a complex problem that requires many services/resources in order to
  - reduce computation time
  - access large databases
  - access special equipments
  - collaborate with other users



Typical Grid application areas

- **High-performance computing (HPC)**
  - to achieve **higher performance** than individual supercomputers/clusters can provide
  - Reguirement: **parallel computing**
- **High-throughput computing (HTC)**
  - To exploit the **spare cycles** of various computers connected by wide area networks
- **Collaborative work**
  - Several users can jointly and remotely solve complex problems

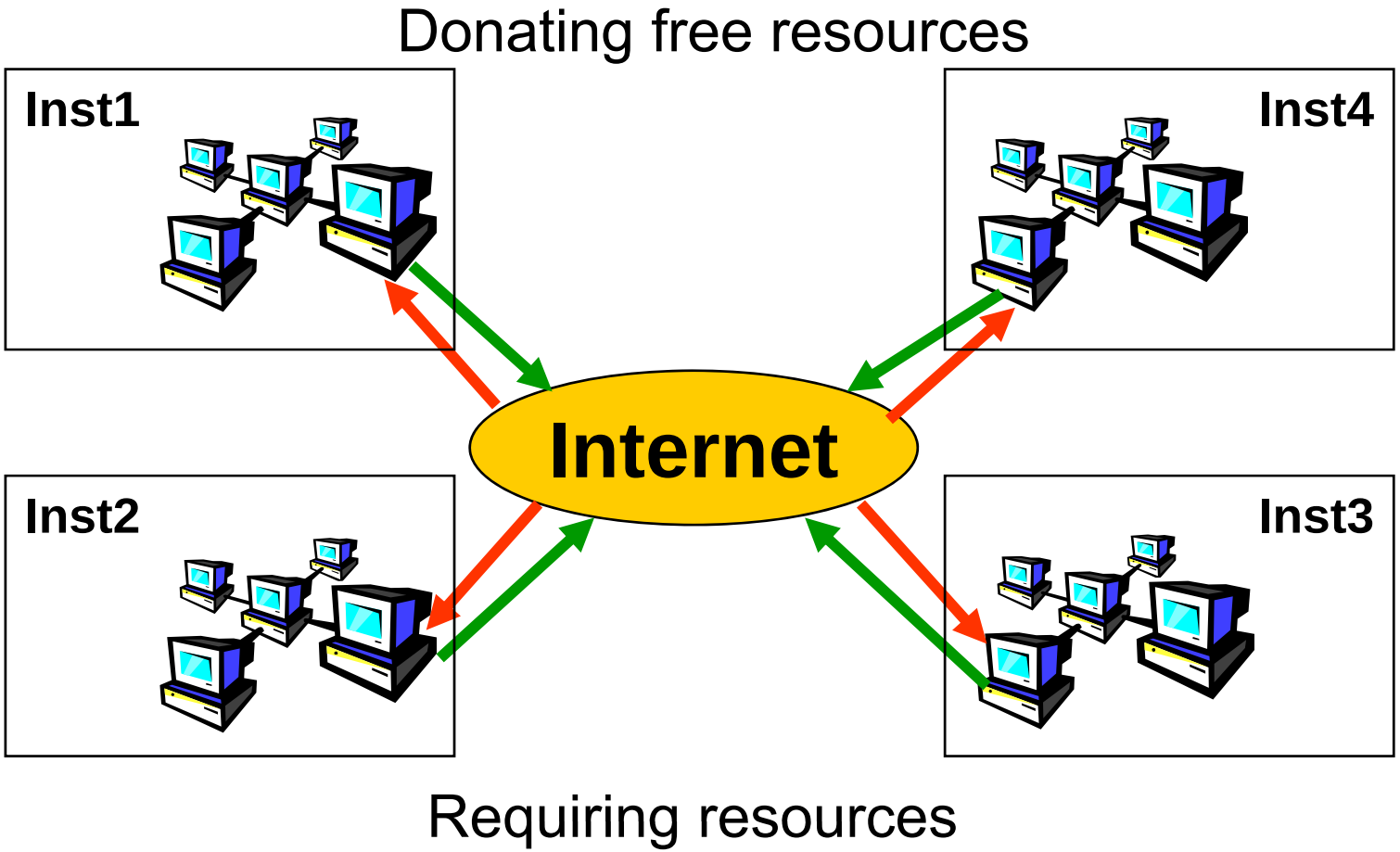
Two basic Grid directions



Two players of the Grid

- Resource donors = D
- Resource users = U
- Relationship between the two characterizes the Grid:
  - if U ~ D => generic Grid model
  - if U >> D => utility Grid model
  - if U << D => desktop Grid model

Generic Grid modell

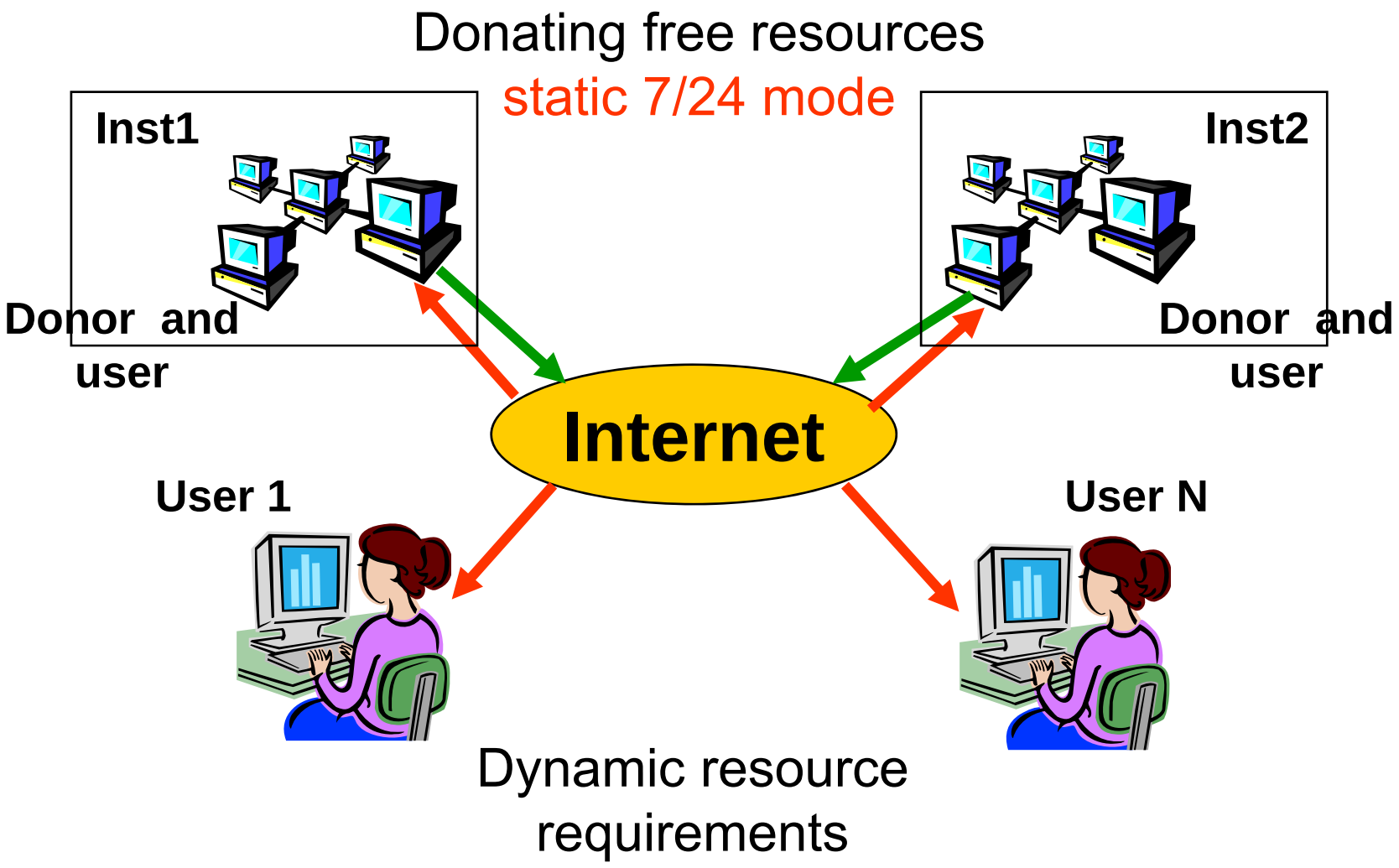


Characteristics of the generic Grid model

- Anybody can donate resources
- Heterogeneous resources, that dynamically join and leave
- Anybody can use the donated resources for solving her/his **own** applications
- Symmetric relationship between donors and users:

  U ~ D
- Examples:
  - GT-2 grids
  - Jini based JGrid (developed in Hungary)
- **Problems:** Installing and maintaining client and server grid software are too complicated

Utility Grid model



Characteristics of the utility Grid model

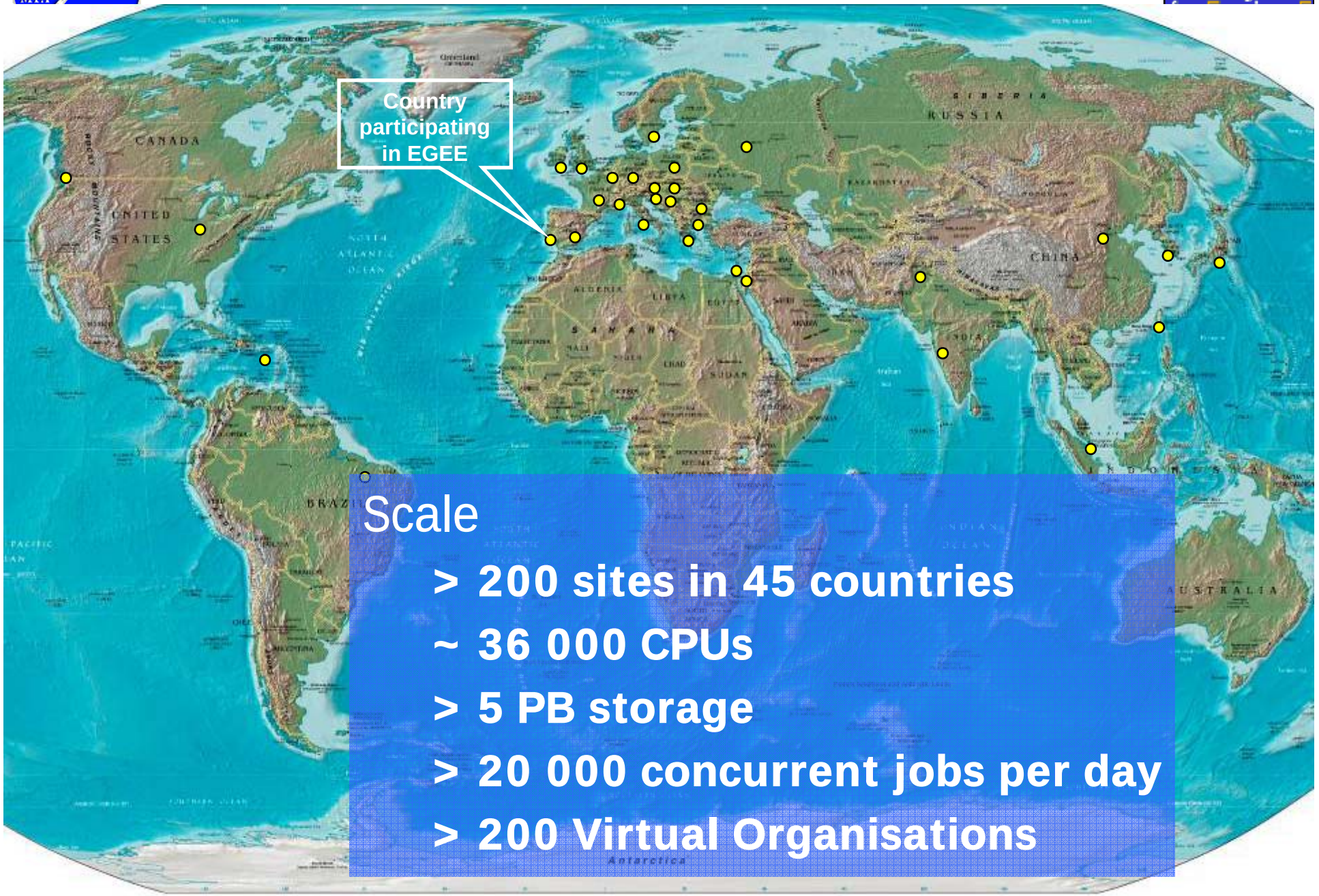
- Donors can be only “professional” resource providers who provide production service (7/24 mode)
- Homogeneous resources
- Anybody can use the donated resources for solving her/his **own** applications
- Asymmetric relationship between donors and users:

  U >> D
- Examples:

  Europe
  - **EGEE → EELA, HunGrid, SEE-Grid**
  - UK National Grid Service (NGS), NorduGrid

  United States of America
  - Open Science Grid (OSG), TeraGrid

The largest production Grid: EGEE



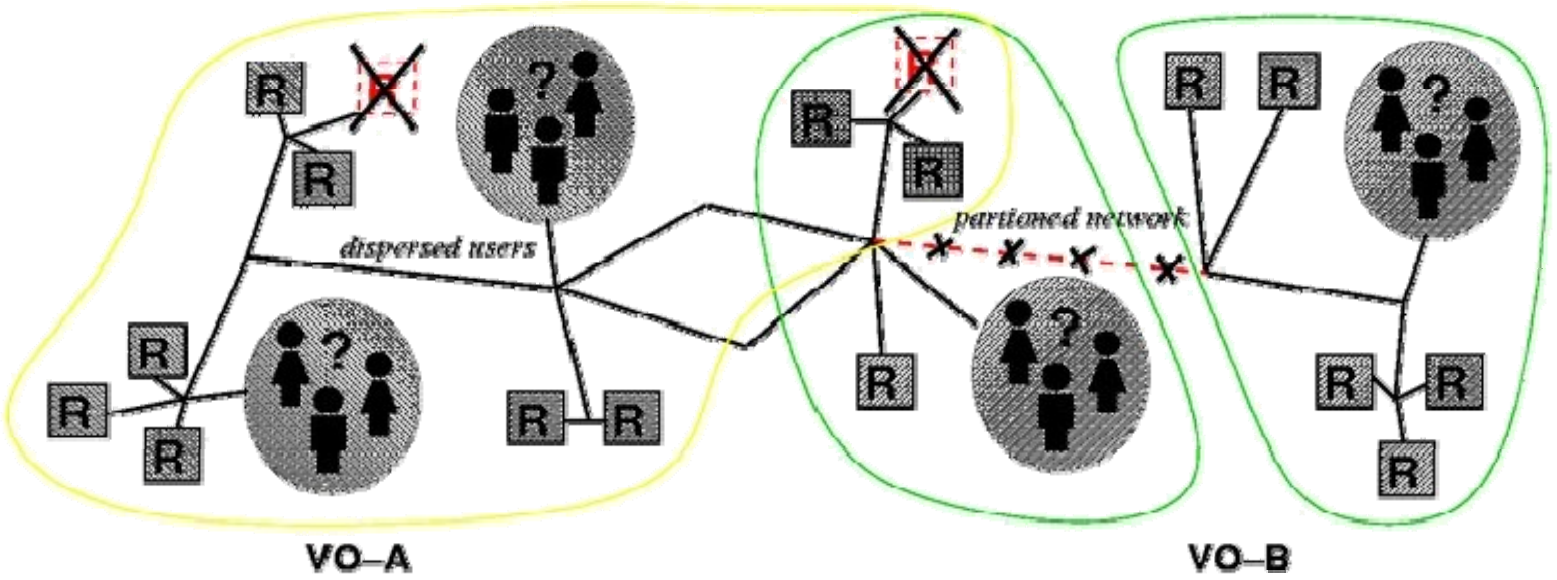
Scale

- **> 200 sites in 45 countries**
- **~ 36 000 CPUs**
- **> 5 PB storage**
- **> 20 000 concurrent jobs per day**
- **> 200 Virtual Organisations**

Utility grids and Virtual Organizations

The Grid problem is to enable "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations."

From **"The Anatomy of the Grid"** by Ian Foster at. al



EGEE and VOs

- EGEE is the largest multi-disciplinary research Grid infrastructure in the world

- EGEE VOs (200+):
  - VO allocated for one specific application/experiment:
    - E.g. LHC VO, ...
  - VO allocated for a research disciplinary:
    - E.g. biomed VO, fusion VO, earth science VO, ...
  - VO allocated for researchers working within the same geographical region:
    - E.g. Hungarian VO (Hungrid), Central European VO (VOCE), ...

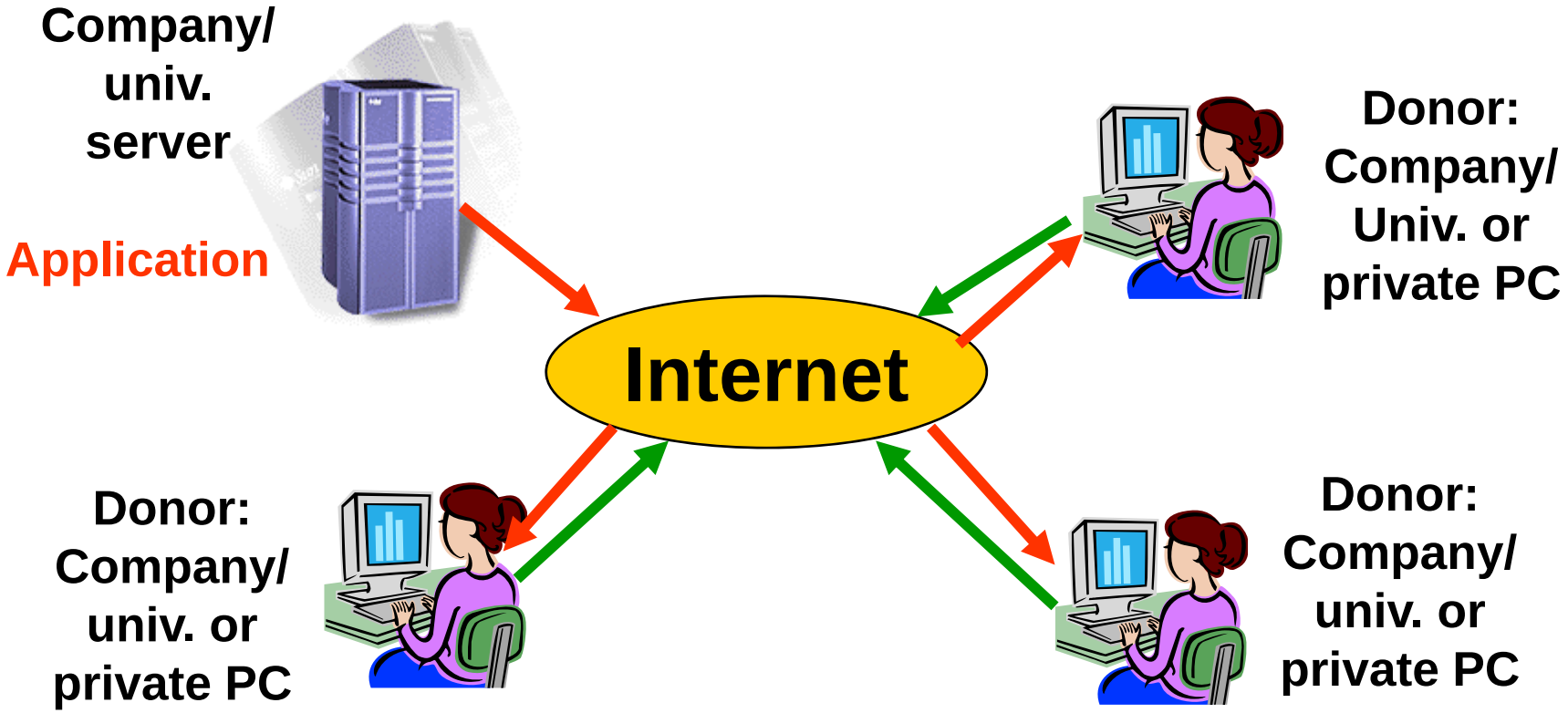
HunGrid: Hungarian VO of EGEE

KFKI-RMKI	SZTAKI	ELTE
▪ 150 processor ▪ 3.4 TB storage	▪ 26 processzor ▪ 2 TB storage	▪ 50 processor ▪ 1.5 TB storage

Additional members: KKKI 12 processor, Veszprémi (6)

Desktop Grid model

Dynamic resource donation



Work package distribution

Characteristics of the desktop Grid model

- Anybody can donate resources
- Heterogeneous resources, that dynamically join and leave
- **One or a small number of projects** can use the resources
- Asymmetric relationship between donors and users:

  U << D

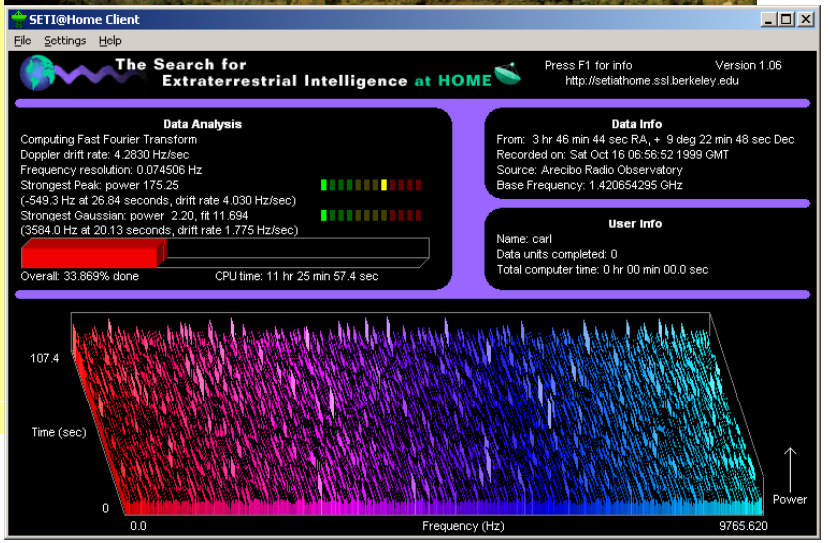
- Advantage:
  - Donating a PC is extremely easy
  - Setting up and maintaining a DG server is much easier than installing the server sw of utility grids

Types of Desktop Grids

- Global Desktop Grid
  - Aim is to collect resources for grand-challenge scientific problems
- Example:
  - BOINC (SETI@home)
- Local Desktop Grid
  - Aim is to enable the quick and easy creation of grid for any community (company, univ. city, etc.) to solve their own applications
- Example:
  - SZTAKI Desktop Grid

SETI: a global desktop grid

- SETI@home
  - 3.8M users in 226 countries
  - 1200 CPU years/day
  - 38 TF sustained (Japanese Earth Simulator is 32 TF sustained)
  - Highly heterogeneous: >77 *different* processor types



SZTAKI Desktop Grid: a local DG system

- Main objective:
  - **Enable the creation of local DG for any community in Hungary**
  - **Demonstrate how to create such a system**
- Three steps to try and use the system:
  1. Donate one PC to test the client site
  2. Port application to the DG server and register PCs for that application
  3. Set up a DG server for the community
- Step 1 is self evident: http://www.lpds.sztaki.hu/desktopgrid/
- SZTAKI helps in steps 2 and 3
- Number of registered donors: 18.000
- Number of registered PCs: 33.000

User concerns of Grid systems

- How to cope with the variety of these Grid systems?
- How to develop/create new Grid applications?
- How to execute Grid applications?
- How to observe the application execution in the Grid?
- How to tackle performance issues?
- How to port legacy applications
  - to Grid systems
  - between Grid systems?
- How to execute Grid applications over several Grids in a transparent way?

Goal of the 2nd Rio Grid School

- **This is a user-oriented and not a grid middleware developer school with goals:**
  - To give answers for the questions above
  - Concentrating of the EGEE Grid technology
  - Teaching the low-level EGEE user interfaces and APIs
  - Showing high-level Grid portal interfaces
  - Showing how to develop applications for the EGEE-like Grids (and for other Grids built with this technology)
  - Introduce the user support systems that EGEE provides