Science gateways

Peter Kacsuk
MTA SZTAKI

Motivations

- There are many user communities who would like to access several DCIs (grids, clouds, clusters) in a transparent way
- They do not want to learn the peculiar features of the used DCIs
- They want to concentrate on their scientific application
- Therefore they need a science gateway

Who are the members of an e-science community?

Science Gateway (SG) Framework Developers (5-10)

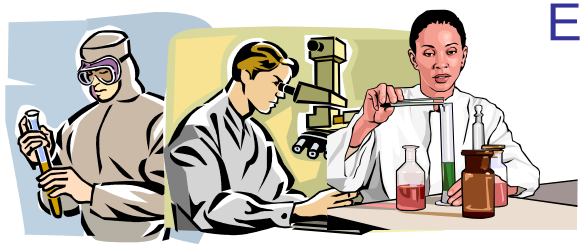
- Develop **generic** SG framework

SG Instance Developers (50-100)

- Develop **application domain specific** SG instance

WF Application Developers (500-1.000)

- Develop WF applications
- Publish the completed WF applications for end-users

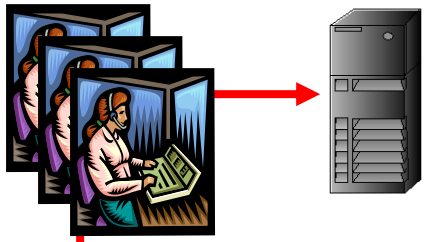


End-users (e-scientists) (50.000-500.000)

- Execute the published WF applications with custom input parameters by creating application instances using the published WF applications as templates

What do WF developers need?

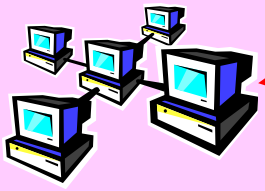
- Using a SG framework to develop applications and share them with others
- Accessing a large set of various DCIs to make these WF applications run



WF App. Repository

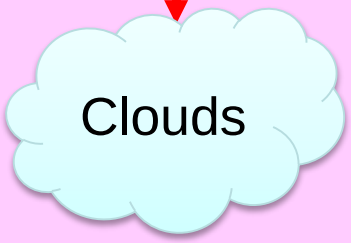
Access to a large set of ready-to-run scientific WF applications

Science Gateway (SG) framework



Local clusters

Supercomputers



Clouds

Supercomputer based SGs (DEISA, TeraGrid)

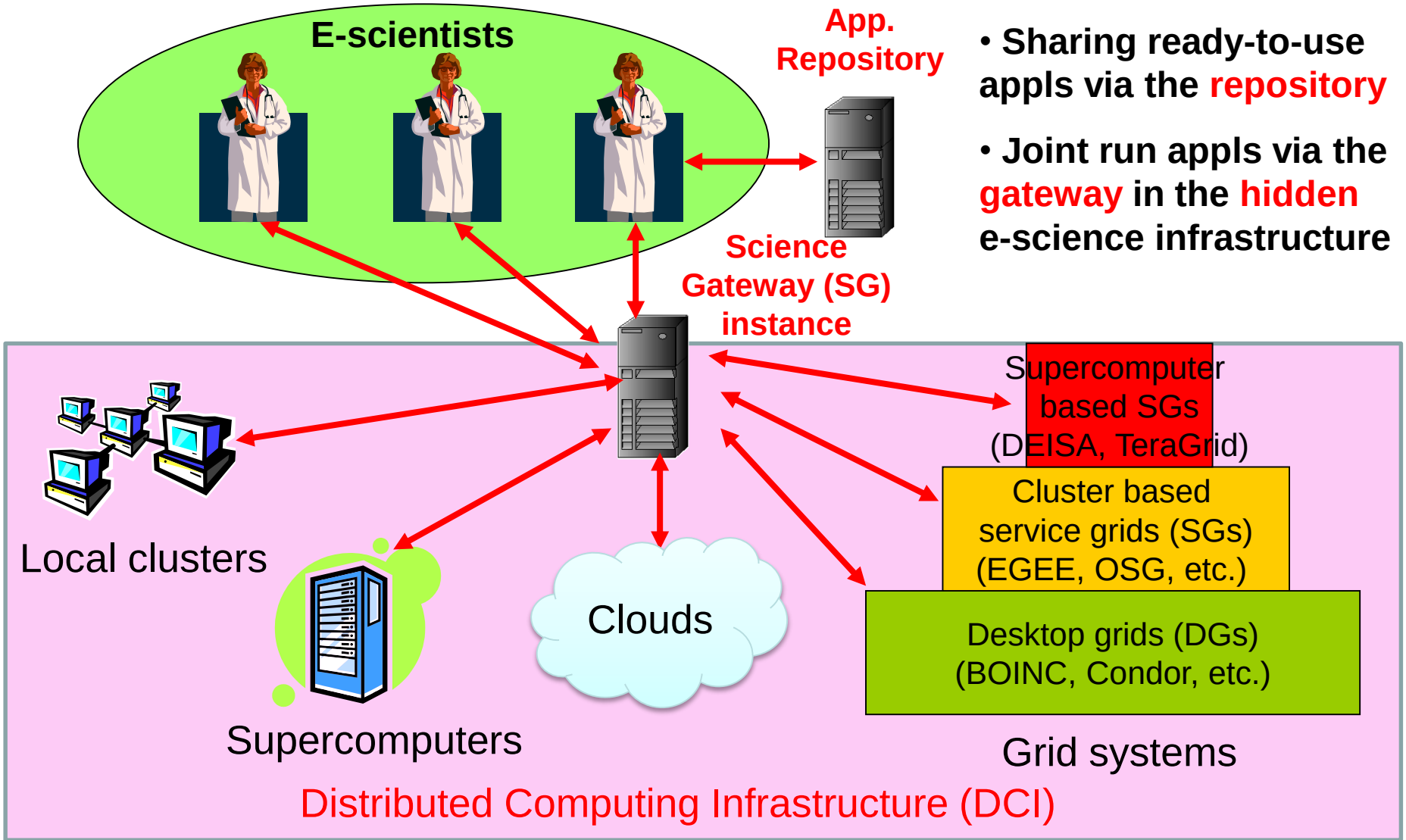
Cluster based service grids (SGs) (EGEE, OSG, etc.)

Desktop grids (DGs) (BOINC, Condor, etc.)

Grid systems

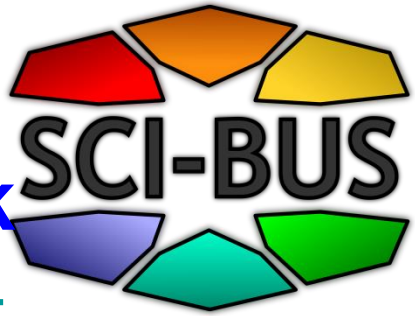
Distributed Computing Infrastructure (DCI)

What do e-scientists need?



- **Sharing ready-to-use appls via the repository**
- **Joint run appls via the gateway in the hidden e-science infrastructure**

WS-PGRADE/gUSE
Generic-purpose gateway framework



- Based on Liferay
- **WS-PGRADE** (Web Services Parallel Grid Runtime and Developer Environment)
- **gUSE** (Grid User Support Environment) architecture
  - General purpose
  - Workflow-oriented portal framework
  - Supports the development and execution of workflow-based applications
  - Enables the multi-DCI execution of any WF
  - Supports the fast development of SG instances by a customization technology

Flexibility of using various DCIs

- Flexible management of **Security:**
  - Individual users' certificate
  - Robot certificates
- Flexible access to **various types of DCIs**:
  - Clusters (PBS, LSF, MOAB, SGE)
  - Cluster grids (ARC, gLite, GT2, GT4, GT5, UNICORE)
  - Supercomputers (e.g. via UNICORE)
  - Desktop grids (BOINC)
  - Clouds (OpenStack, OpenNebula, Amazon)

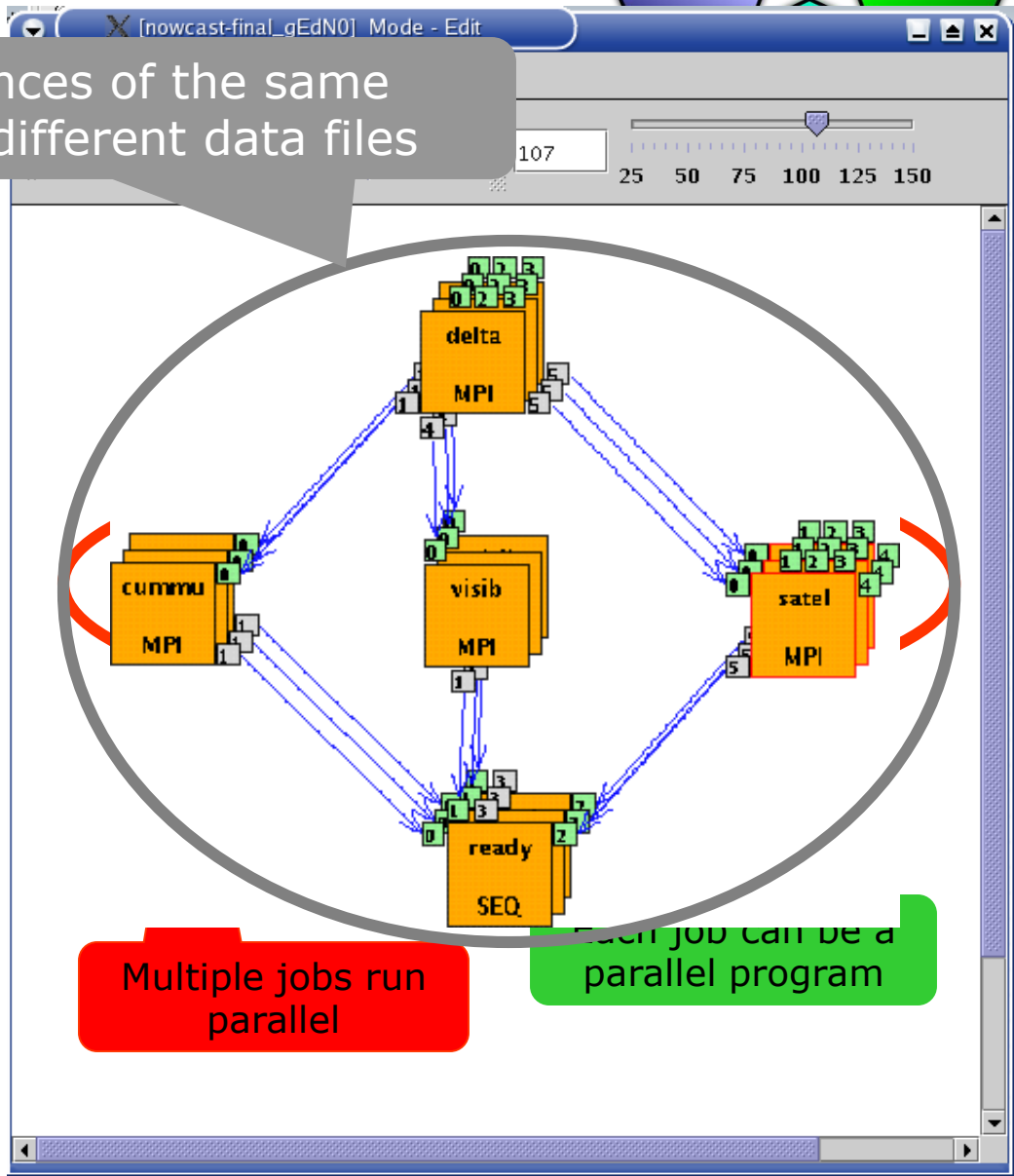
Flexibility in exploiting parallelism

Multiple instances of the same workflow with different data files

– Parallel execution inside a workflow node

– Parallel execution among workflow nodes

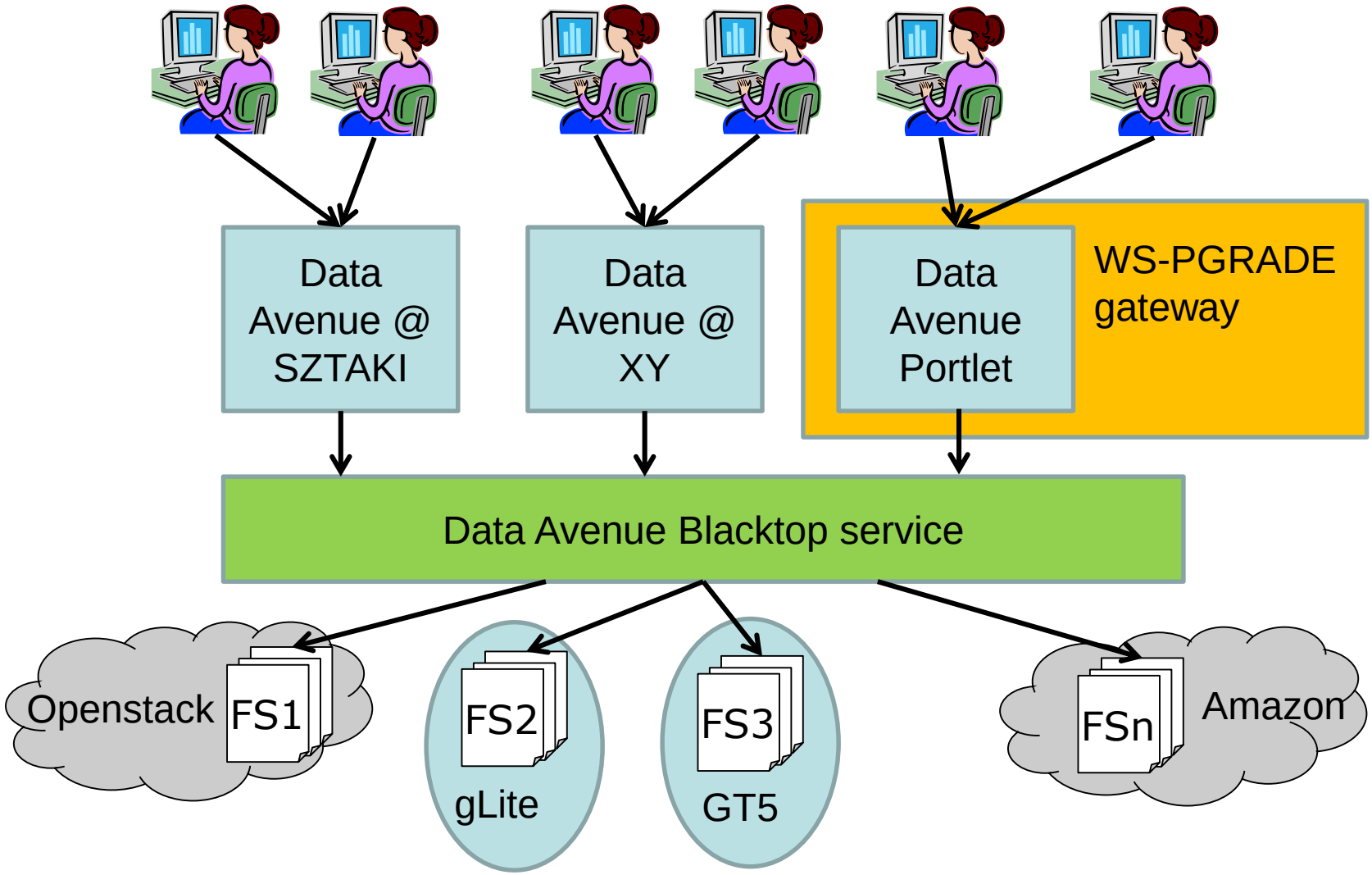
– Parameter study execution of the workflow



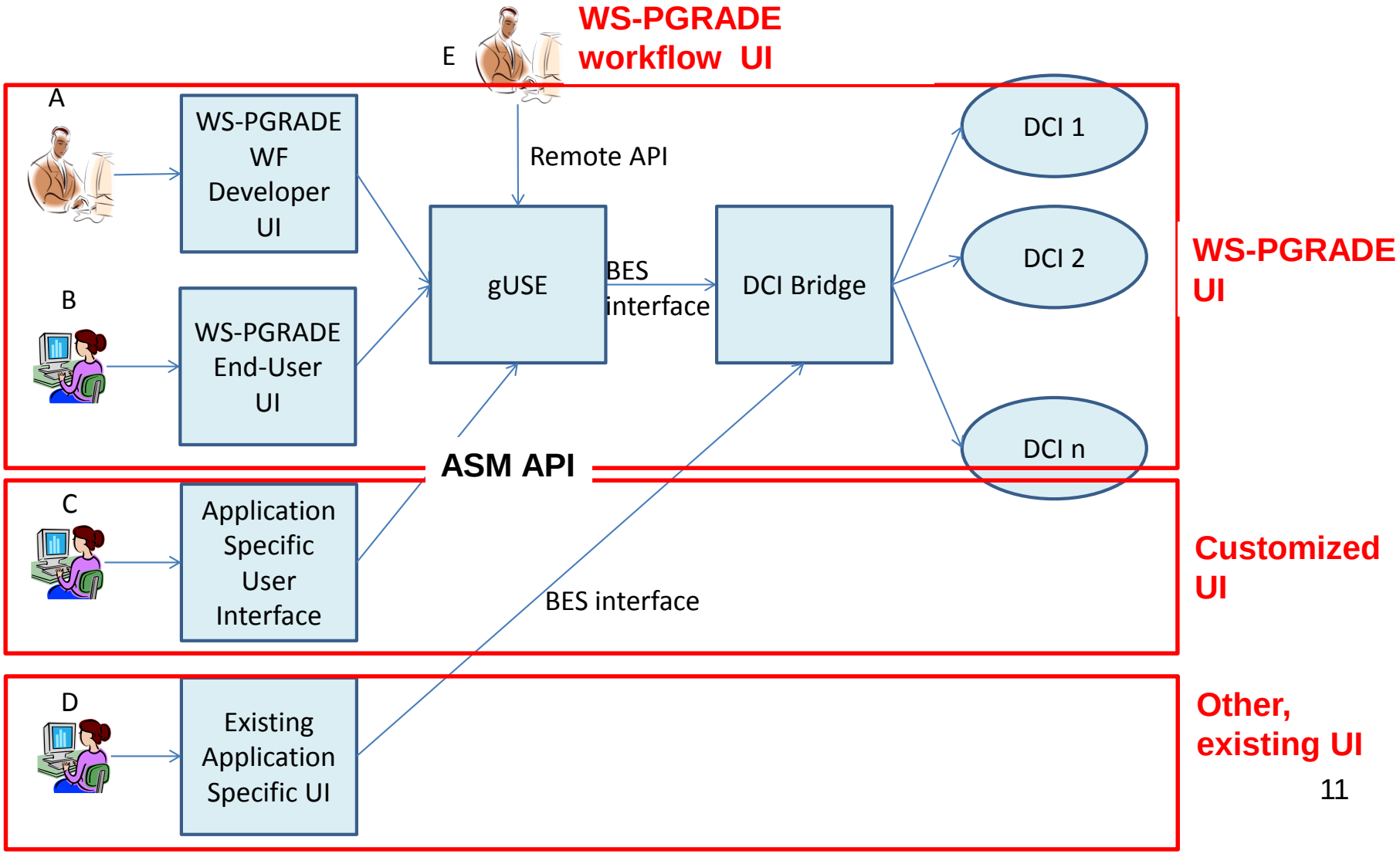
Flexibility in data storage access

- **Use Data Avenue Blacktop service**
  - To access data storages in different DCIs
  - To transfer files among the storages of different DCIs
  - To upload/download files to/from the storages of different DCIs
- **Data Avenue Liferay portlet** to access the data transfer services of Data Avenue Blacktop
- See details: http://data-avenue.eu/home
- Currently supported protocols:
  - http, https, ftp, gsiftp, srm, iRODS, S3

Data Avenue services



Flexibility in user access modes



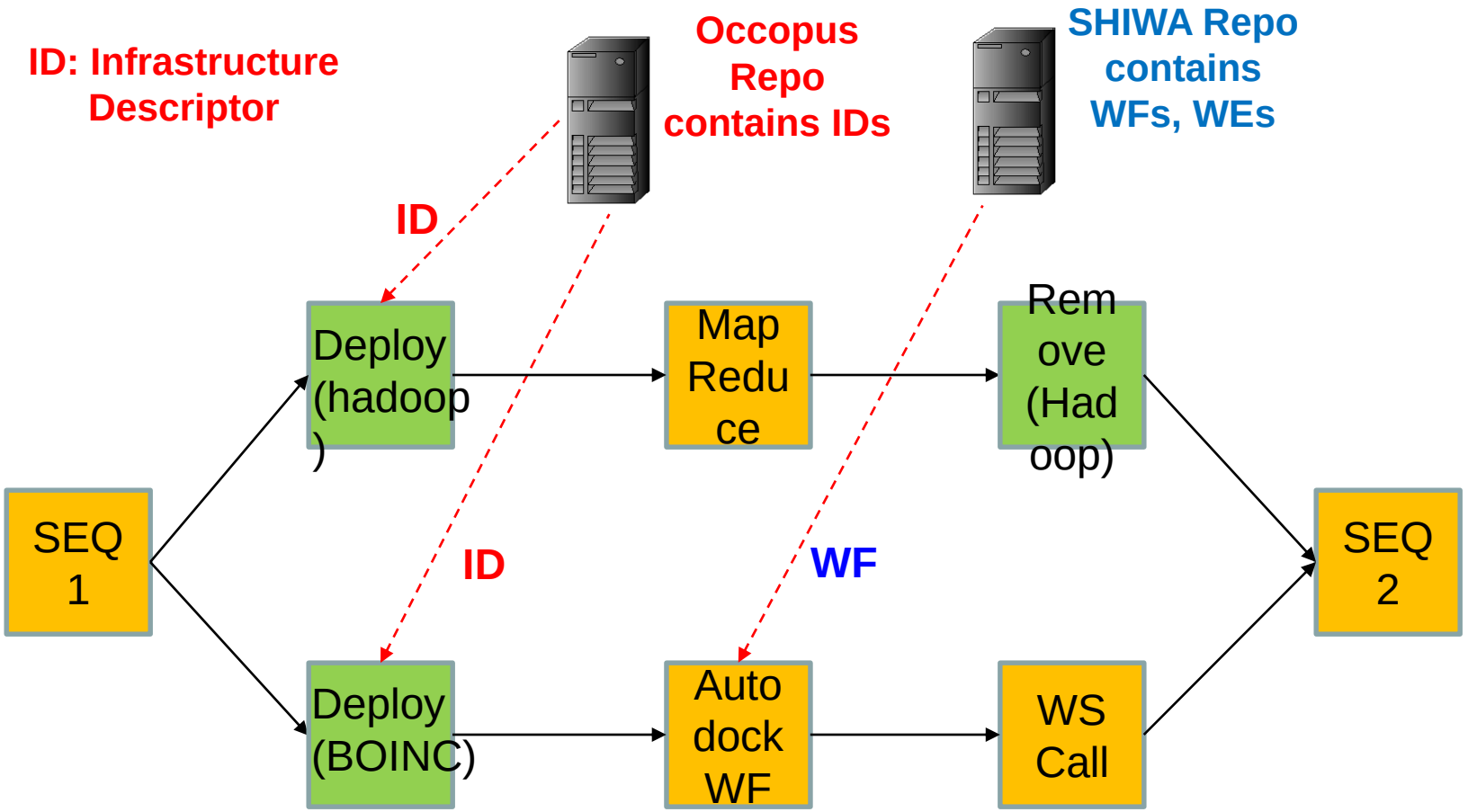
Flexibility in science gateways types

1. Generic purpose gateways for national grids
   - Core WS-PGRADE/gUSE (e.g. Greek NGI, MTA Cloud in Hungary)
2. Generic purpose gateway for a particular DCI
   - EDGI gateway based on WS-PGRADE/gUSE
3. Application-oriented science gateway instance
   - MoSGrid gateway
   - VERCE gateway
   - VIALACTEA gateway
4. Generic purpose gateway for specific technologies
   - SHIWA gateway for workflow sharing and interoperation

Further development plans

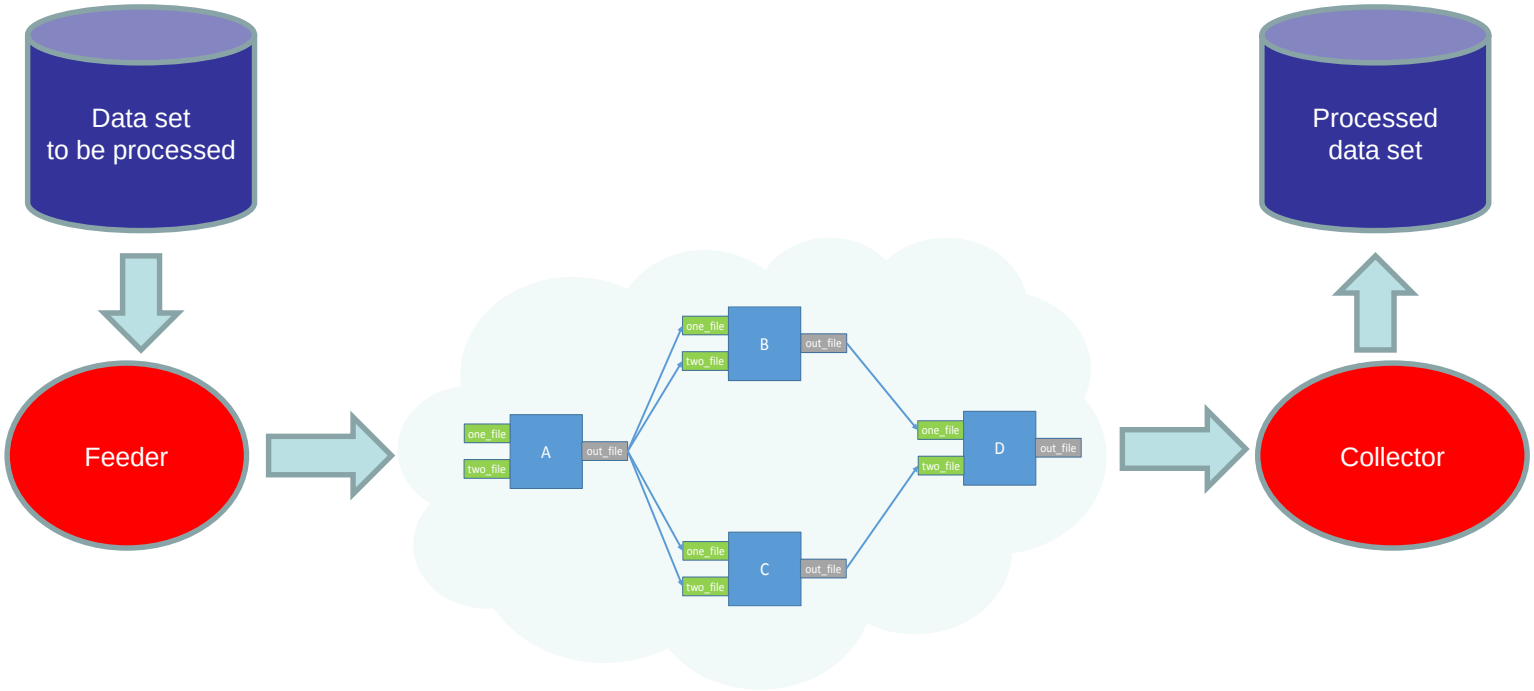
- Integrating with the Occopus cloud orchestrator tool:
  - Mapreduce/Hadoop portlet
- Infrastructure-aware workflow concept:
  - You can define the required infrastructure in the cloud and Occopus will deploy it. Examples:
    - Hadoop cluster
    - Docker cluster
    - MICADO scalable application framework (COLA project)

Infrastructure-aware workflow

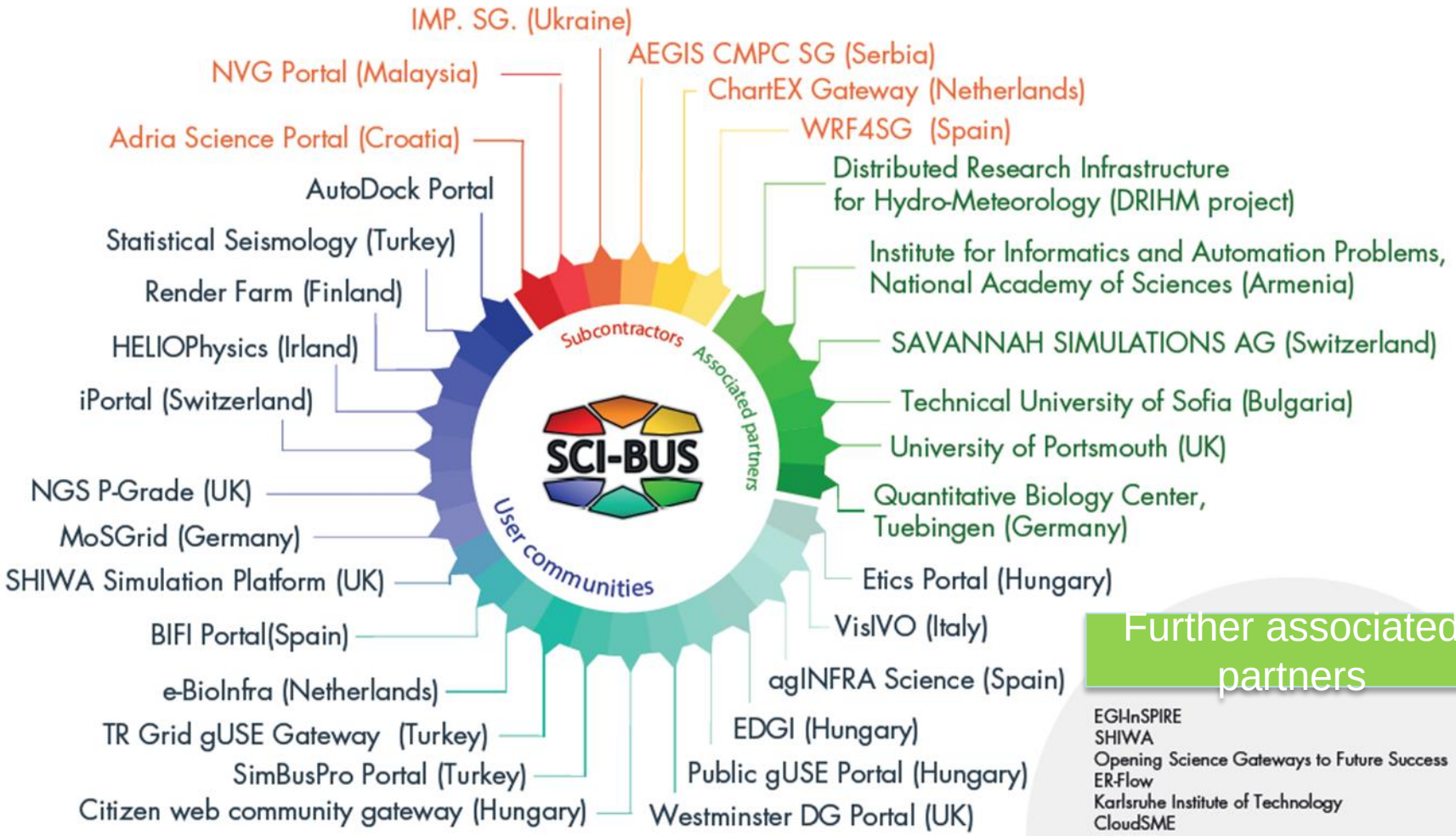


Further development plans

- Enhancing with data stream oriented workflow:
  - Flowbster (based on Occopus)



Gateway solutions



Further associated partners

EGHnSPIRE
SHIWA
Opening Science Gateways to Future Success
ER-Flow
Karlsruhe Institute of Technology
CloudSME
IDGF-SP
DRIHM
VERCE

http://www.sci-bus.eu/science-gateways

sourceforge and gUSE based gateways

- More than 100 deployments world-wide
- More than 23.000 downloads from 85 countries on sourceforge

Conclusions

- If you want to develop a SG instance, use a SG framework instead of developing it from scratch
- Why to select WS-PGRADE/gUSE?

1.Robustness

- Already large number of gateways used in production

2.Sustainability

- Although the SCI-BUS project is over still there are projects that maintain or further develop this technology. Most important: **EOSC-hub**

3.Functionalities

- Rich flexible functionalities

4.How easy to adapt for the needs of a new user community?

- Already large number of gateways customized from gUSE/WS-PGRADE