www.egi.eu
@EGI_eInfra

Turkish Science E-Infrastructure

Capabilities & Ideas for Integration

Hakan Bayındır
TUBITAK ULAKBIM

The work of the EGI Foundation
is partly funded by the European Commission under H2020 Framework Programme

In This Presentation

- Turkish Science E-Infrastructure (TRUBA)
  - Short History
  - Capabilities
- Ideas for Integration

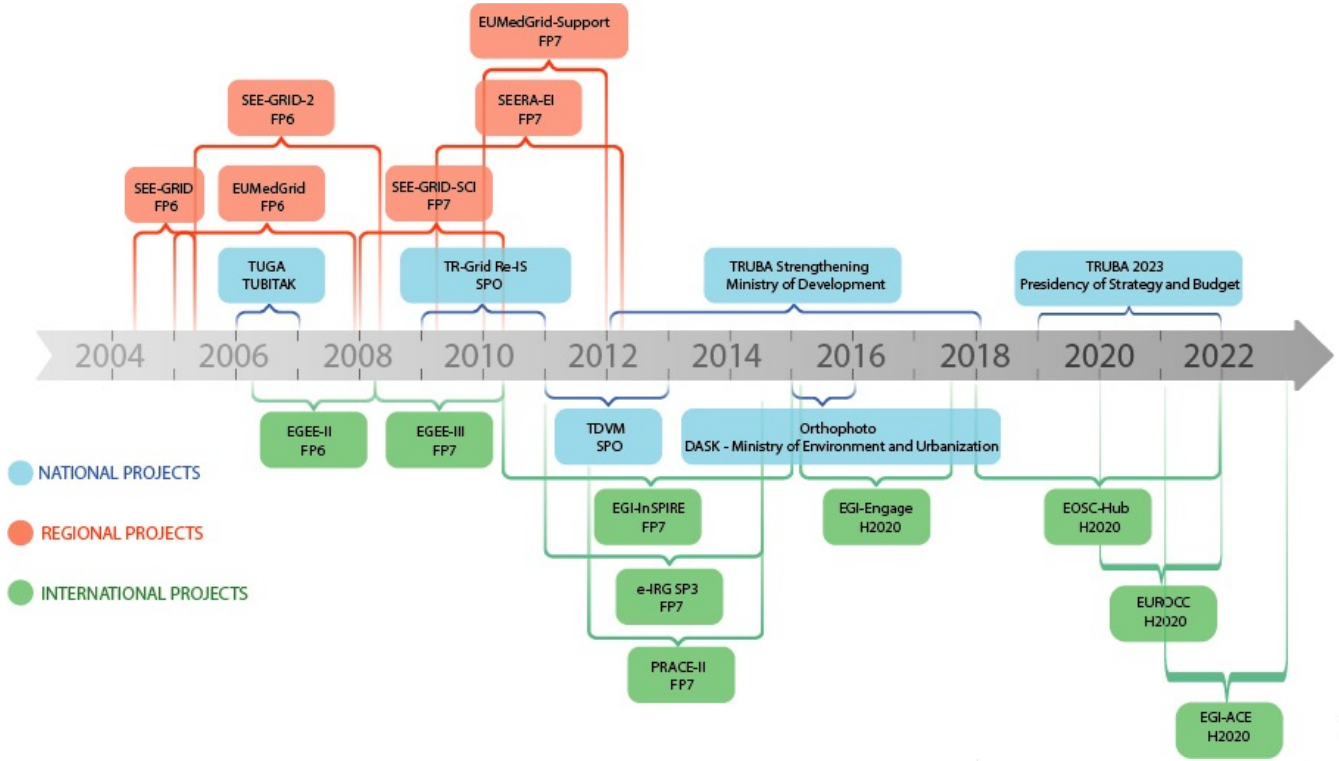
Turkish Science E-Infrastructure

A Short History

- Started in 2003 under TR-Grid name and renamed as TRUBA in 2010.
- Operating HPC clusters since 2003 and Grid sites since ~2004.
- Participated in many FP6 and FP7 projects prior to EGI.
- EGI Cloud Compute Provider in EOSC.
- EGI-ACE project partner.

Turkish Science E-Infrastructure

A Short History



Turkish Science E-Infrastructure

Capacities and Capabilities

HPC cluster

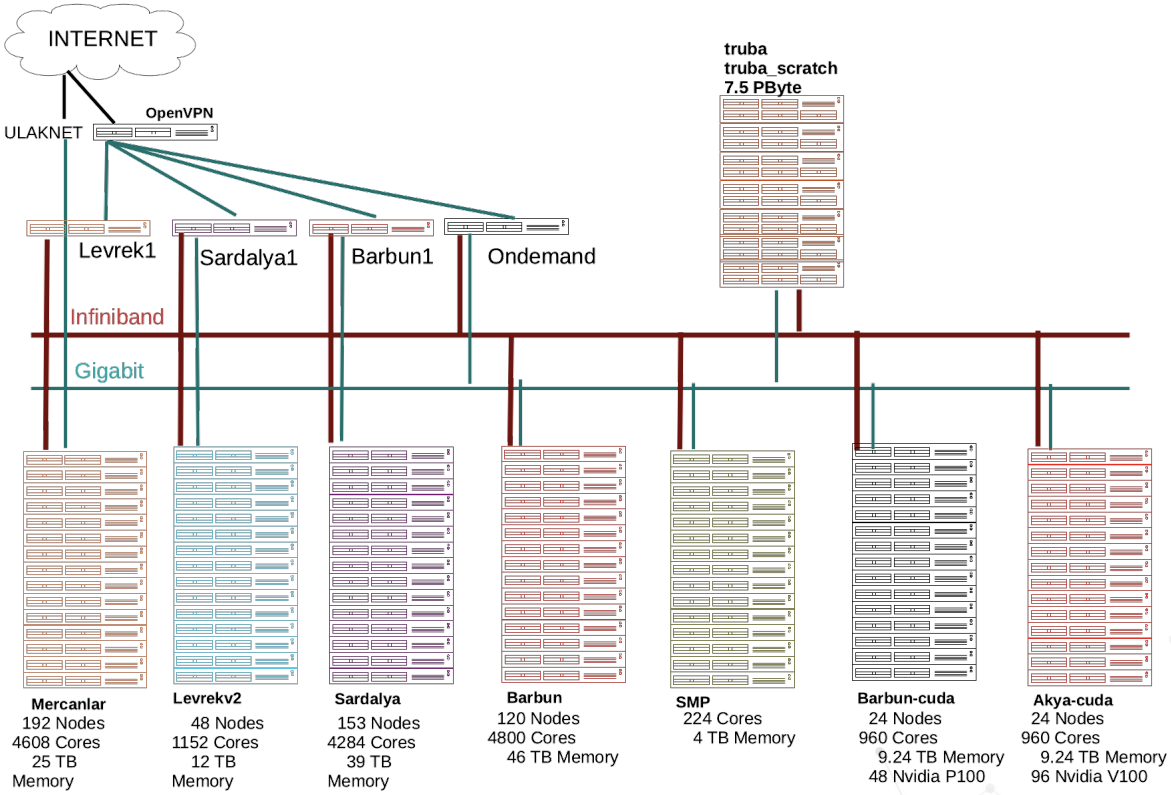
- 13M CPU Hours / Month
- ~20.000 CPU Cores connected with InfiniBand.
- 48 Tesla P100 + 96 Tesla V100 GPUs
- 14PB of storage, powered by Lustre FS.
- > 3000 registered, > 350 active users.

Grid

- 2 Grid sites, 1 FedCloud site.
- 2.5PB Tier-2 storage, >800TB FedCloud storage.

Turkish Science E-Infrastructure

HPC Cluster Architecture



Turkish Science E-Infrastructure

More on OnDemand

- OnDemand is a simple web front end for HPC systems.
  - Allows SSH, file management.
  - Provides job designer, builder and submitter.
  - Can monitor submitted jobs.
  - Provides desktop environment and desktop application access.
    - Some applications (Jupyter, R Studio, Tensorboard, etc.) are already supported.
- Used by TRUBA to ease users' workflows.
- Maybe usable in our scenario.
  - No work has been done to test/adapt OnDemand by TUBITAK.

HPC Integration with Cloud

Integration Challenges & Ideas

- Many areas of concern:
  - Authentication & authorization
  - Data ingest and export
  - Workflow orchestration & monitoring
  - Accounting
- Many levels of abstractions
  - Authentication & Access
  - Job submission and monitoring
  - Containers?

HPC Integration with Cloud

A theroretical workflow

1. Authenticate the user and submit the job to the cluster
2. Get the code and run the job
    1. Get the data
    2. Do the work
    3. Send results back
    4. Clean the environment
3. Notify the user

Looks simple, but has a lot of gaps.

HPC Integration with Cloud

Questions in mind

- **User authentication:** From where, with which mechanism?
- **User mapping to HPC:** Pool accounts?
- **Job design & submission:** Defining job specifications, converting to scheduler format.
- **Data & Job:** Getting the code and data to site, data retention, user mapping retention.
- **Monitoring & accounting:** Letting user know job state, aggregating usage stats.

HPC Integration with Cloud

Some Ideas & Open Questions

- Using standard “utility” containers may help with data, setup and clean-up.
  - Bring data in according to a manifest.
  - Send results after job is done
  - Clean (if required) before leaving.
- Accounting is job monitoring is relatively easy.
  - Interface a simple daemon with job scheduler.
- Authentication, job design and submission is hard
  - Very dependent on access and authentication design.

EGI: Advanced Computing for Research

www.egi.eu
@EGI_eInfra

Thank you
for your attention.

Questions?

The work of the EGI Foundation
is partly funded by the European Commission under H2020 Framework Programme