

# A SIMPLE API FOR GRID AND DISTRIBUTED APPLICATIONS. (SAGA)

---

Danila Oleynik

01.06.2013 ASCR meeting.

# Introduction

- SAGA (*Simple API for Grid Applications*) defines a high-level interface to the most commonly used distributed computing functionality. SAGA provides an access-layer and mechanisms for distributed infrastructure components like job schedulers, file transfer and resource provisioning services
- Behind the API facade, SAGA-Python implements a flexible *adaptor* architecture. Adaptors are dynamically loadable modules that interface the API with different middleware systems and services.
- <http://saga-project.github.io>

# Supported Middleware

- **Job Submission Systems**

*(All queuing system adaptors can also access clusters remotely by tunneling commands through SSH and GSISSH)*

- SSH and GSISSH
- Condor and Condor-G
- PBS and Torque
- Sun Grid Engine
- SLURM

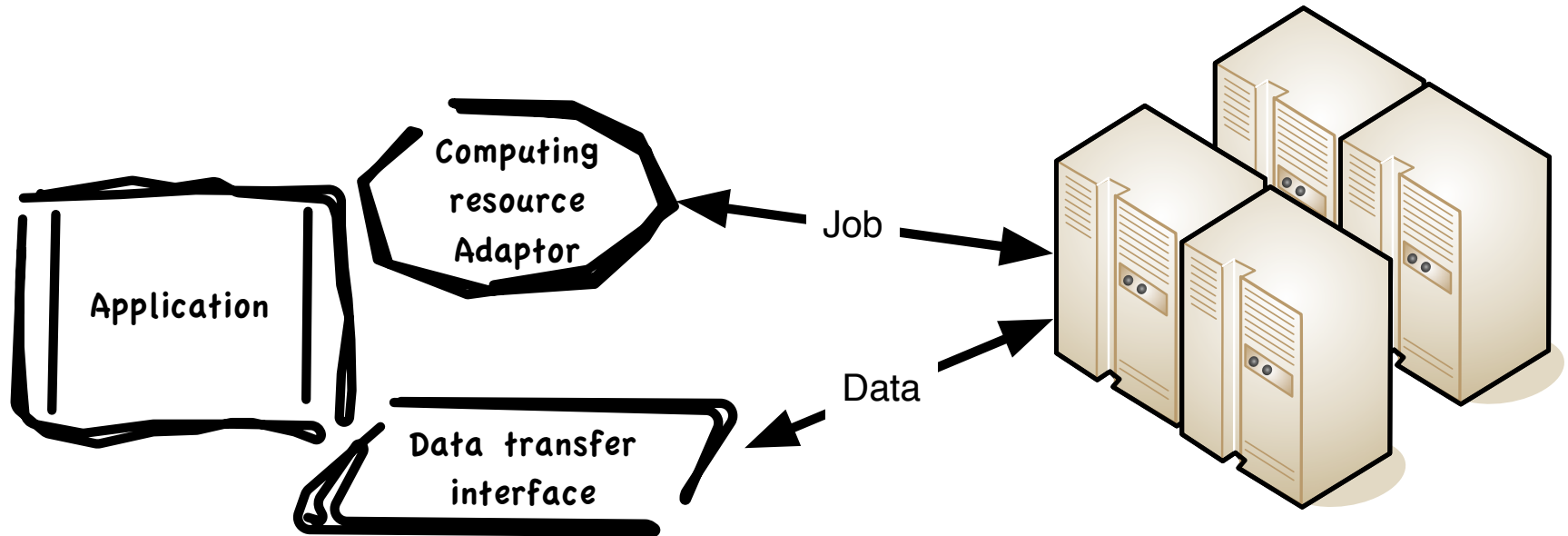
- **File / Data Management**

- SFTP/GSIFTP
- HTTP/HTTPS

- **Resource Management / Clouds**

- EC2 (libcloud)

# How it works



**Computing Resource:**  
Condor and Condor-G,  
PBS and Torque,  
Sun Grid Engine,  
SLURM,  
Remote host (SSH)

# Other features

- Configuration and logging
- Arguments and environment
- Special parameters for batch processing systems
- Different security contexts:
  - UserPass
  - SSH
  - X.509
  - MyProxy
  - EC2
  - EC2\_KEYPAIR
- And much more other features (not all features still documented 😊 )

# Some experience

- Configuration:
  - SAGA application host: laptop
  - Job processing: remote hosts in UTA, JINR, Ixplus
  - Adaptor: SSH
  - Security context: UserPass, SSH RSA
- Results
  - In most of cases simple test like 'Hello world' passed
    - Authorization on Ixplus with RSA got some troubles.

# Next step

- Some more complicated tests: upload executable to remote host, collecting parameters of execution.
- Testing with batch systems: PBS(Torque), SGE, Condor
  - Agreed to use Tier3 simulating infrastructure in JINR for this tests.