



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Programming Distributed Computing Platforms with COMPSs

Workflows & Distributed Computing Group

28-29/01/2019

Barcelona

COMPSs Execution Environments



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

COMPSs Architecture

Python App

C/C++ App

Java App



How can I select the execution platform?



Grid



Cluster



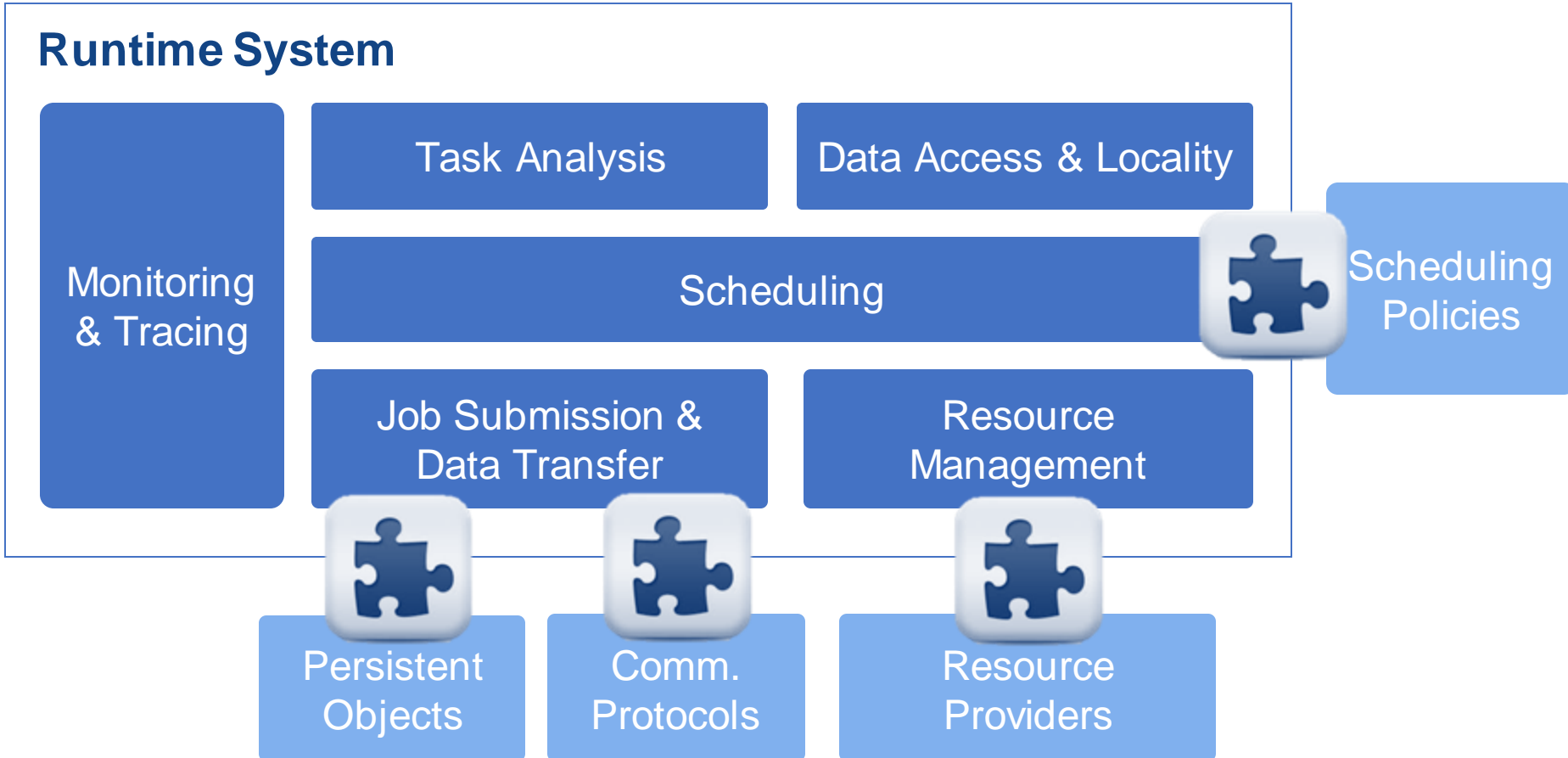
Cloud



Containers



Runtime Extensions



Execution Environments Configuration

Specifies:

- Resources and Project files
- Scheduler, Comm. Adaptor
- Persistent object storage



runcomps options

Infrastructure Description

- Describe the available resource in the infrastructure
- Describe Cloud Providers: Images and VM Templates

Runtime System

Exec. Mngmt & Data Transfers

Persistent Object Storage

DataClay

Hecuba

Redis

Communication Adaptor

NIO

GAT

Schedulers

Resources

Cloud Connector

jClouds

rOCCI

Slurm

Docker

Mesos

resources.xml

project.xml

Master-Worker Comm. Mechanism

- GAT: Restricted environments (only ssh access) and Grid Middleware
- NIO: Efficient Persistent workers implementation in controlled and secured environments

Resource Scalability

- Provide interaction with resource providers to create and destroy new computing resources

Application Exec. Desc.

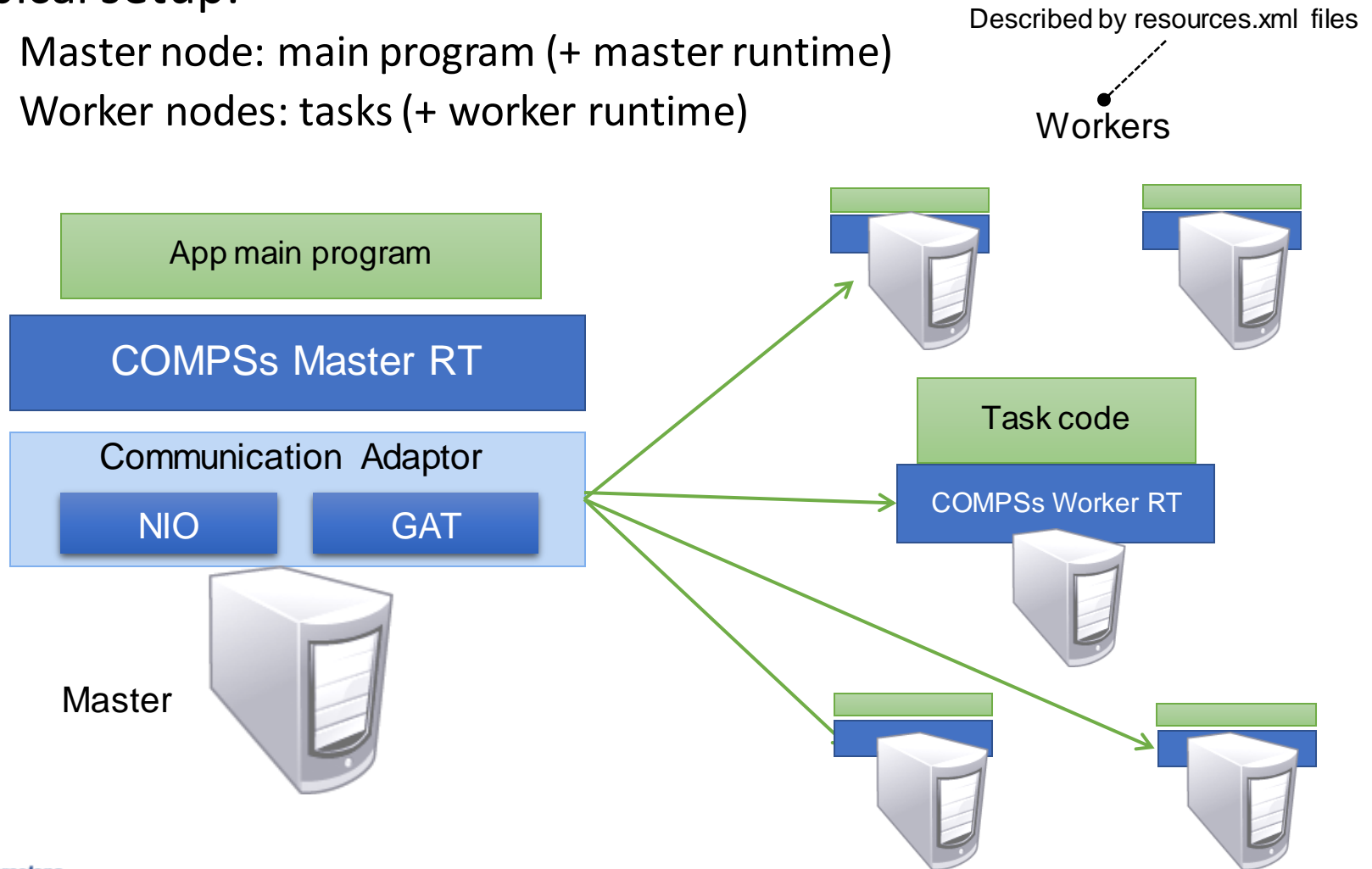
- Selection of resources
- Application Code Location
- Working directory
- Provided as execution command argument

Basic Execution Environments

- Interactive Computing Nodes
- Clusters (interaction with batch jobs systems)
- Clouds (interaction with Cloud Provider APIs)

COMPSs @ Interactive Hosts

- Typical setup:
 - Master node: main program (+ master runtime)
 - Worker nodes: tasks (+ worker runtime)



Configuration: Resources Specification

- Resources.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ResourceList>
  <!--Description for any physical node-->
  <ComputeNode Name="172.20.200.18">
    <Processor Name="P1">
      <ComputingUnits>4</ComputingUnits>
      <Architecture>amd64</Architecture>
      <Speed>3.0</Speed>
    </Processor>
    <Memory>
      <Size>256.2</Size>
      <Type>Non-volatile</Type>
    </Memory>
    <Storage>
      <Size>2000.0</Size>
    </Storage>
    <OperatingSystem>
      <Type>Linux</Type>
      <Distribution>OpenSUSE</Distribution>
      <Version>13.2</Version>
    </OperatingSystem>
    ...
  </ComputeNode>

```

```

    <Software>
      <Application>Java</Application>
      <Application>Python</Application>
    </Software>
    <Adaptors>
      <Adaptor Name="integratedtoolkit.nio.master.NIOAdaptor">
        <SubmissionSystem>
          <Interactive/>
        </SubmissionSystem>
        <Ports>
          <MinPort>43001</MinPort>
          <MaxPort>43002</MaxPort>
        </Ports>
      </Adaptor>
    </Adaptors>
  </ComputeNode>

  <ComputeNode Name="172.20.200.19">
    ...
  </ComputeNode>
</ResourceList>

```


Configuration: Project Specification

- Project.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Project>
  <!--Description of used nodes in an application and where is the application installed-->
  <ComputeNode Name="172.20.200.18">
    <InstallDir>/opt/COMPSS/</InstallDir>
    <WorkingDir>/tmp/</WorkingDir>
    <Application>
      <AppDir>/home/user/apps/app_A/</AppDir>
      <LibraryPath>/home/user/apps/app_A/lib</LibraryPath>
      <Classpath>/home/user/apps/app_A/clases/</Classpath>
      <Pythonpath>/home/uthser/apps/app_A/clases/py<Pythonpath>
    </Application>
  </ComputeNode>

  <ComputeNode Name="172.20.200.19">
    ...
  </ComputeNode>
  ....
</Project>
```

COMPSs@Cluster

- Execution divided in two phases
 - Launch scripts queue a whole COMPSs app execution
 - Actual execution starts when reservation is obtained

Cluster Login Node



enqueue_comps

Automatically generated XML files

Queue System (LSF, PBS, ...)

Cluster Compute Nodes

Application

COMPSs RT

Communication Adaptor

NIO



COMPSs@MN

Cluster Login Node



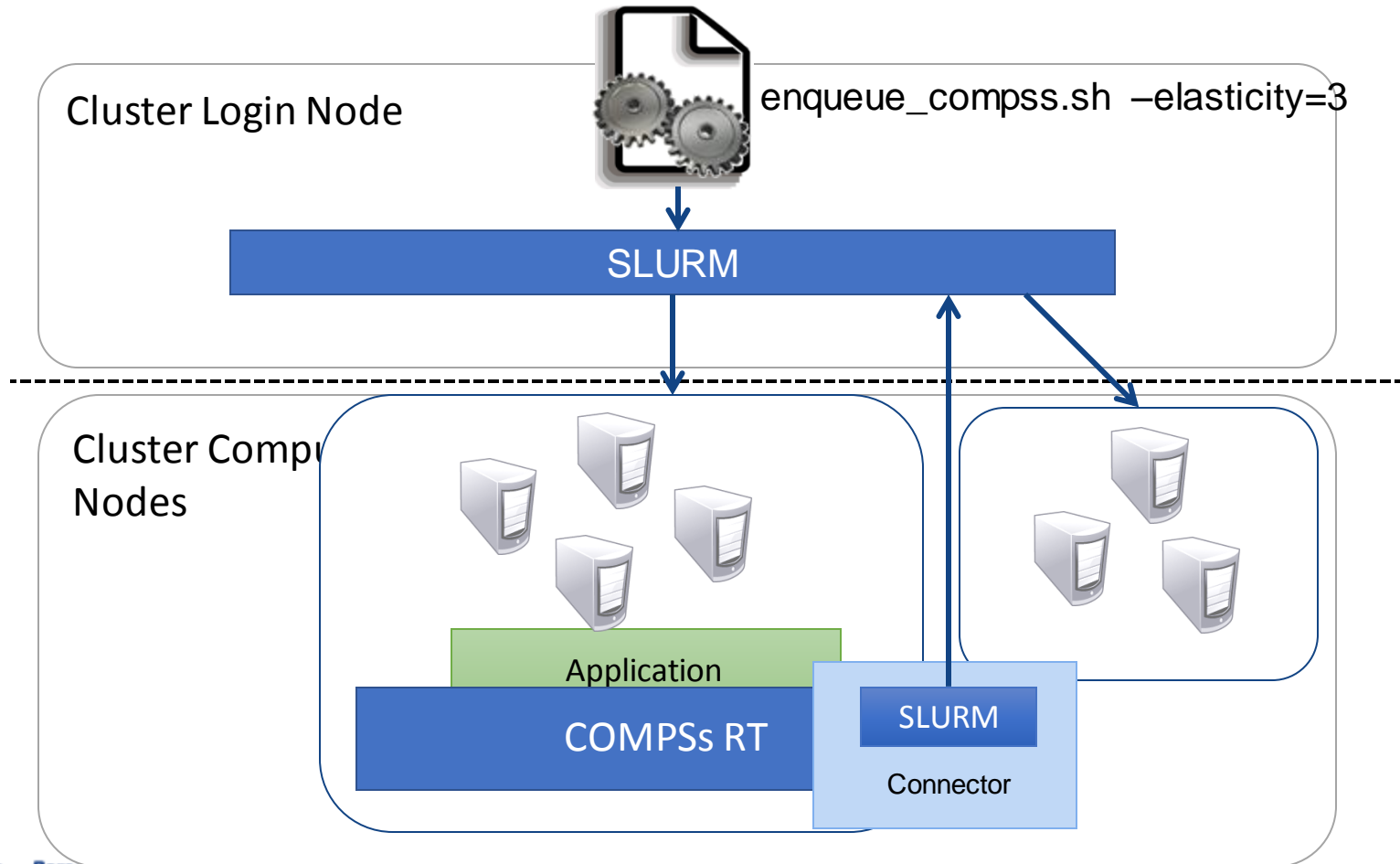
enqueue_comps

Next Hands-on!!



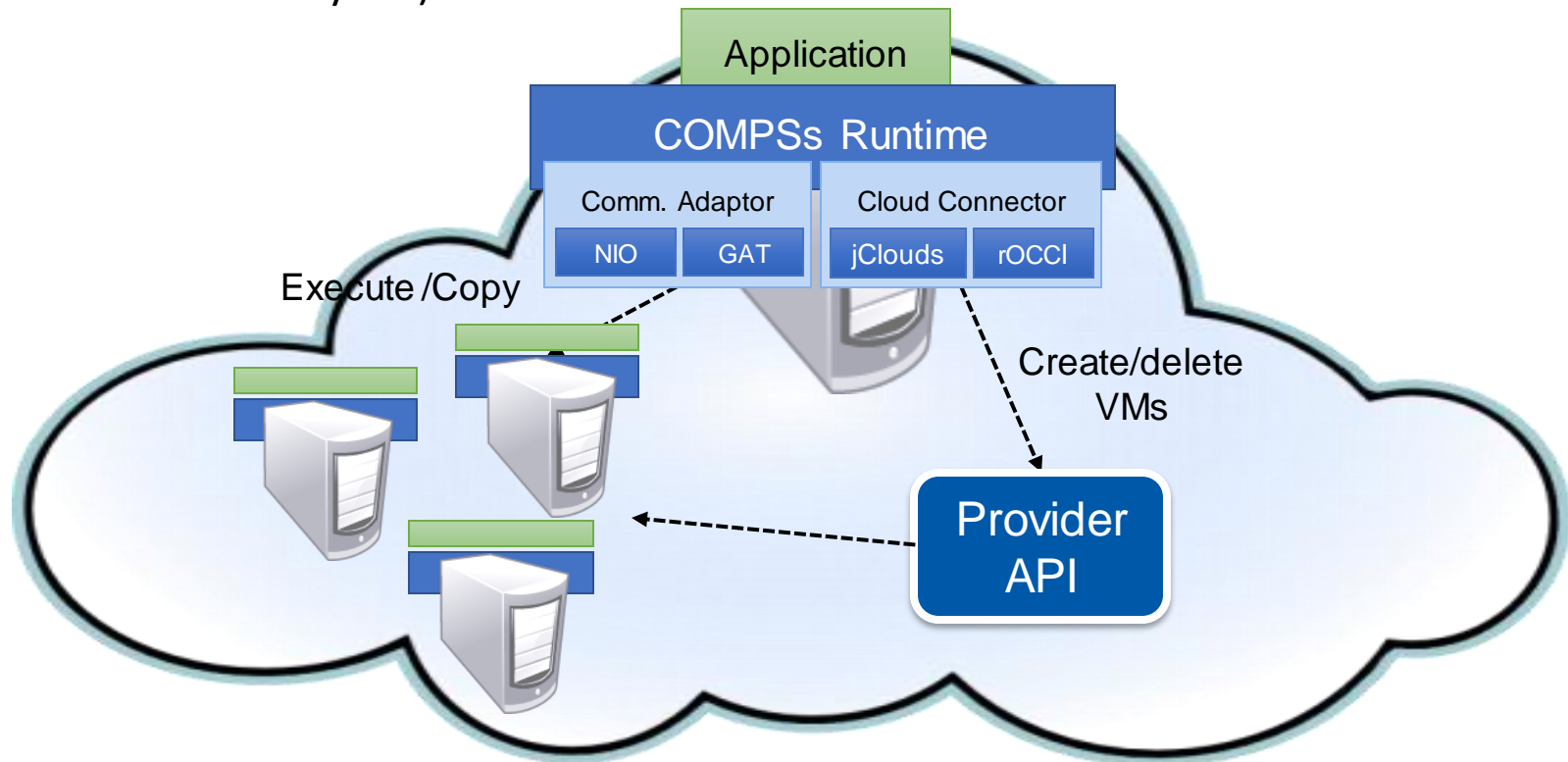
Elasticity@Clusters with SLURM Connector

- Enable the SLURM connector at submission time



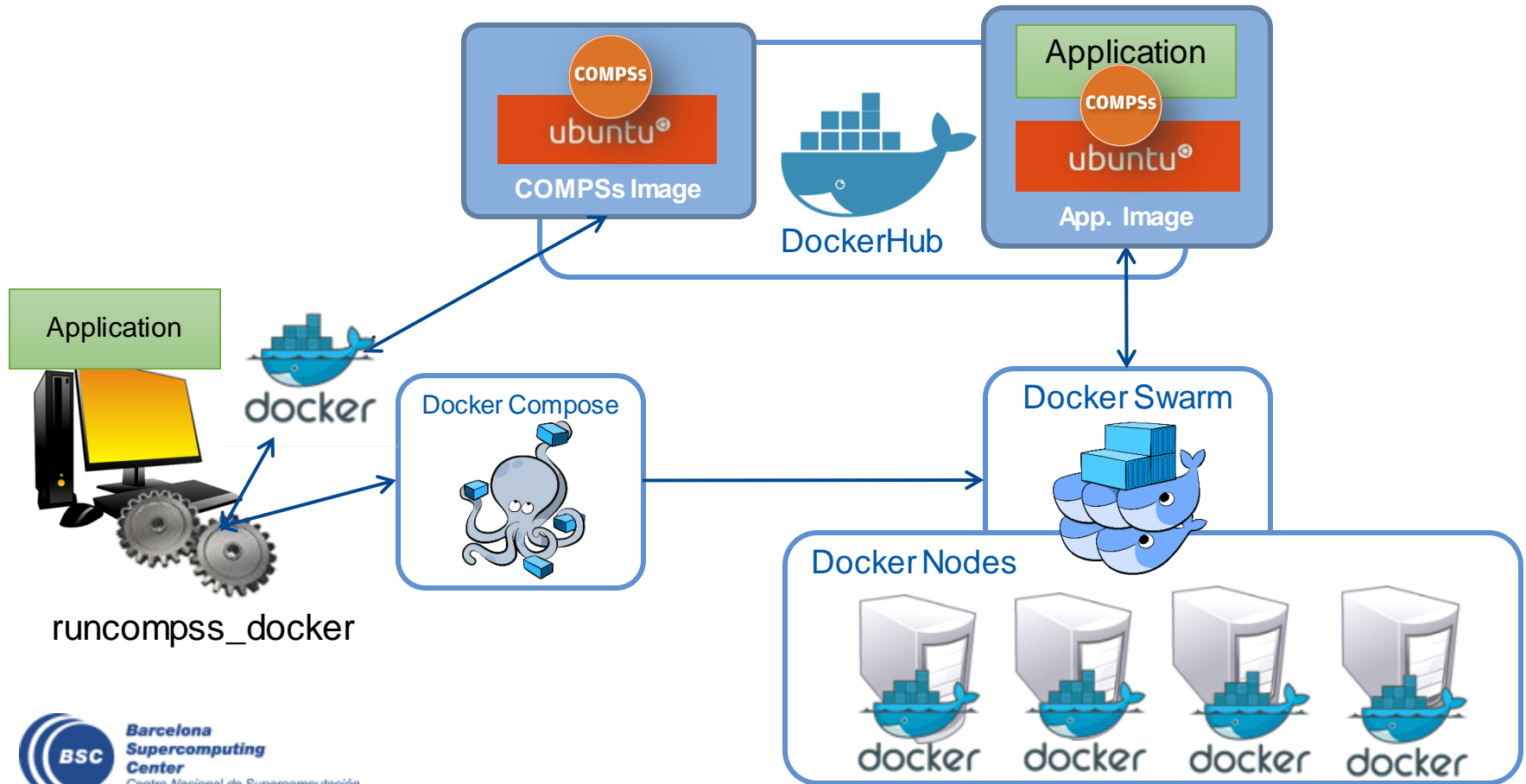
COMPSs@Cloud

- Execution of COMPSs applications in Clouds
 - Select the connector to interact with the Cloud provider
 - Adaptor to communicate VMs (NIO if provider supports firewall management, GAT if only ssh)



COMPSs@Docker

- Keep as transparent for the user as possible
 - Same as running a local COMPSs application (runcompss command)
- Deploy applications as a set of docker container



COMPSs@Singularity

- Execute applications from a container image in HPC cluster
- Can be also used in combination with the cluster elasticity

