

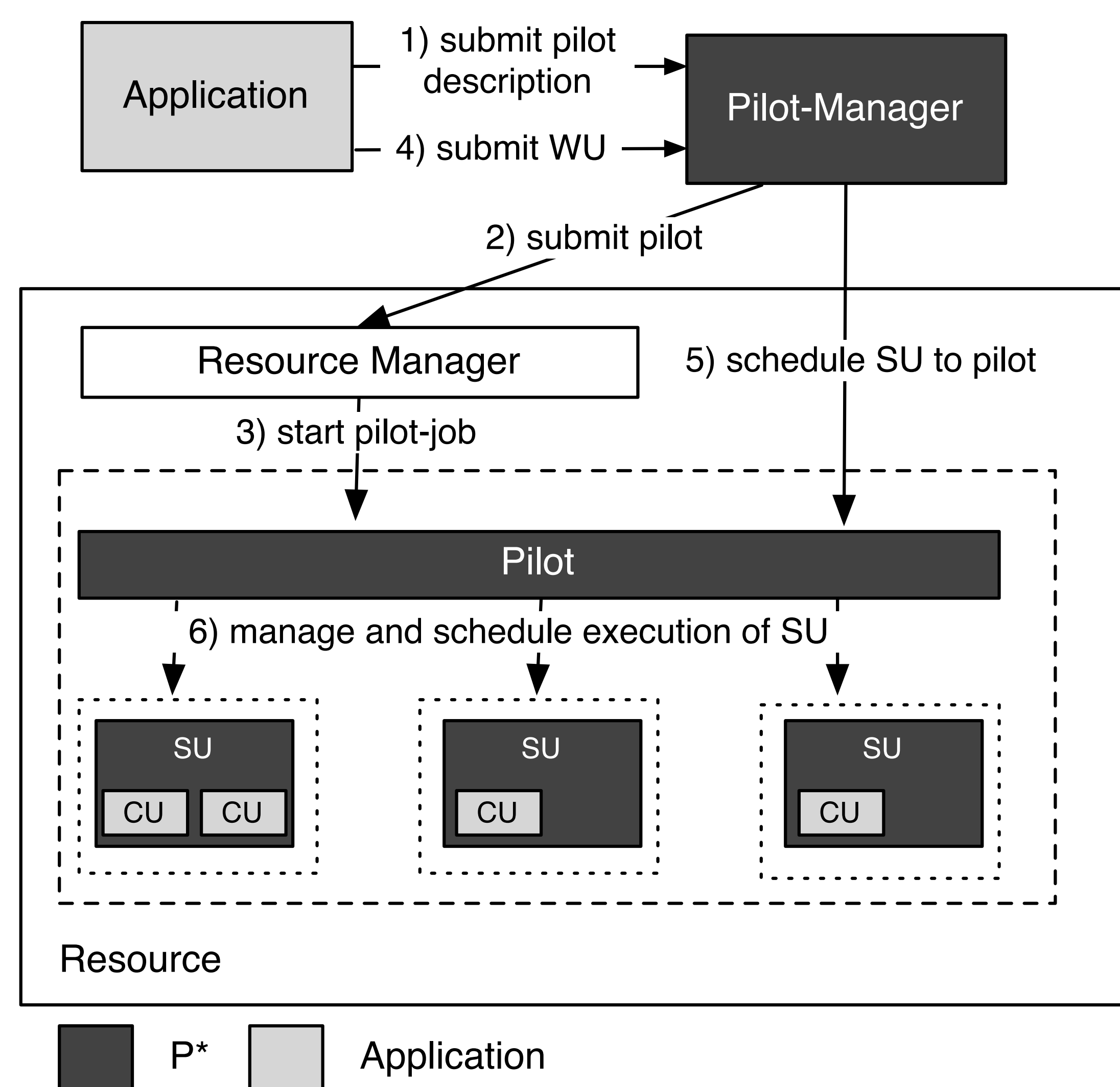
# P\*: Towards a Common Model for Pilot-Jobs

Andre Luckow, Mark Santcroos, Ole Weidner, Andre Merzky, Sharath Maddineni, Shantenu Jha

## Pilot-Jobs

Pilot-Jobs (PJ) have become one of the most successful abstractions in distributed computing. In spite of extensive uptake, there does not exist a well defined, unifying conceptual model of Pilot-Jobs, which can be used to define, compare and contrast PJ implementations. This presents a barrier to extensibility and interoperability. This work is an attempt to, provide a minimal but complete model (P\*) of Pilot-Jobs and to demonstrate the interoperable and concurrent usage of distinct pilot-job frameworks on different production distributed cyberinfrastructures via the use of an extensible API for the P\* Model (Pilot-API).

Pilot-Jobs provide an effective abstraction for dynamic execution and resource utilization. As a consequence of providing a simple approach for decoupling workload management and resource assignment/scheduling, Pilot-Jobs have been one of the most successful abstractions in distributed computing.



## P\* Model

The **P\* Model** provides a minimal, but complete model of Pilot-Jobs and provides a conceptual basis to compare and contrast different PJ frameworks. The P\* Model comprises of the following elements:

**Pilot:** The pilot is the entity that actually gets submitted and scheduled on a resource. The PJ provides application (user) level control and management of the set of allocated resources.

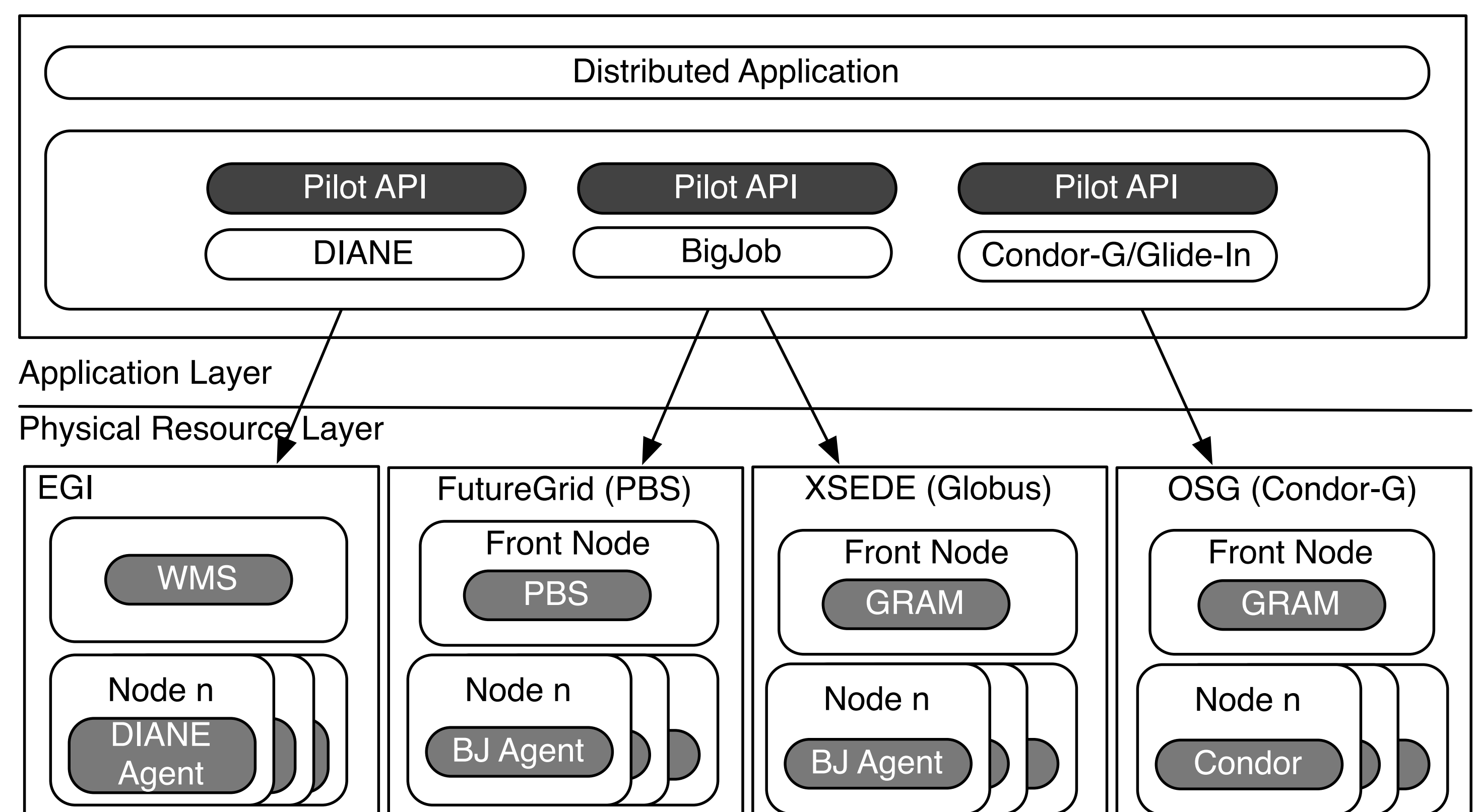
**Compute Unit (CU):** A CU encapsulates a self-contained piece of work ( a task) specified by the application that is submitted to the Pilot-Job framework. There is no intrinsic notion of resource associated with a CU.

**Scheduling Unit (SU):** SUs are the units of scheduling internal to the P\* Model, i.e., it is not known by or visible to an application.

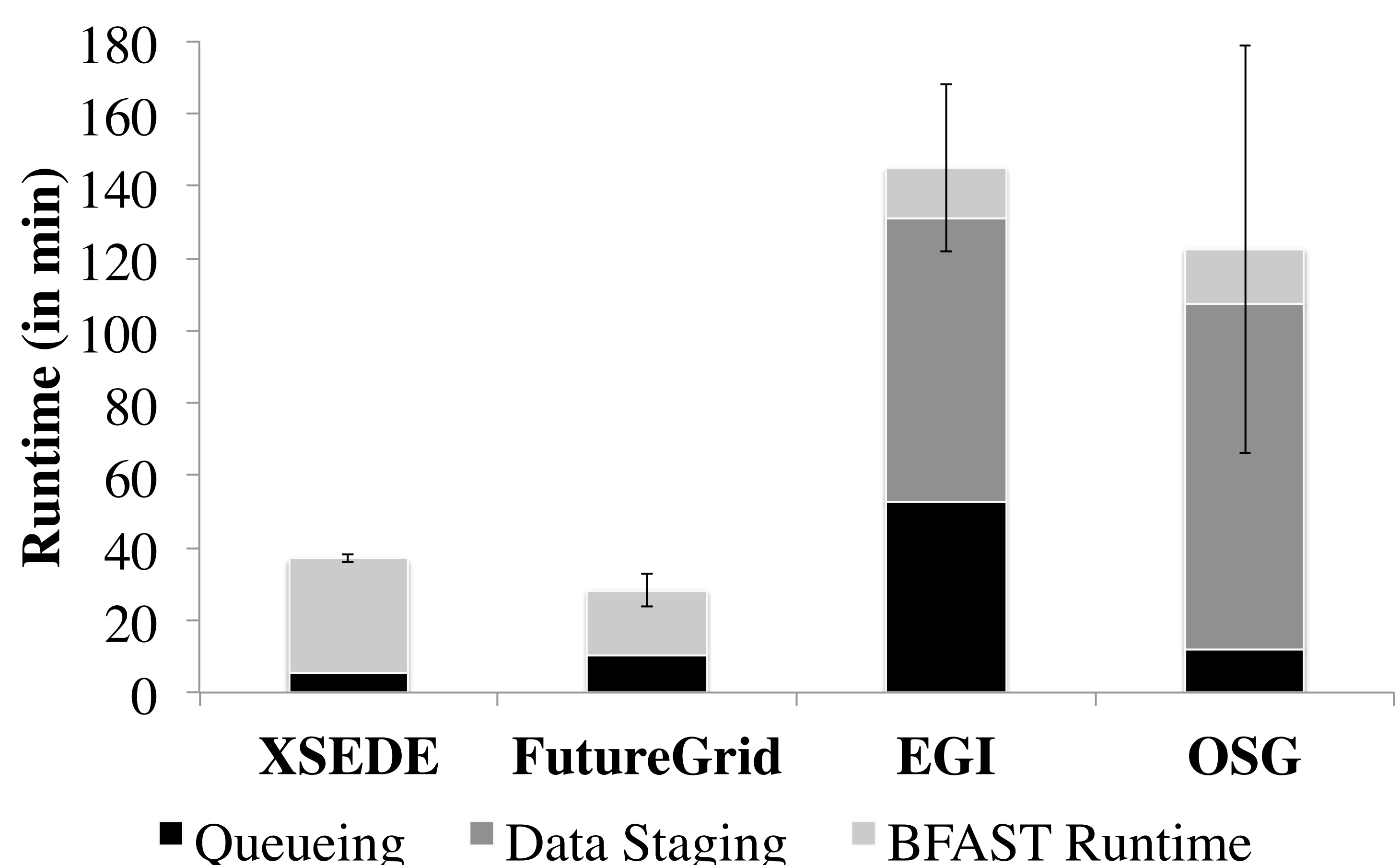
**Pilot-Manager (PM):** The PM is responsible for (i) orchestrating the interaction between the Pilots as well as the different components of the P\* Model (CUs, SUs) and (ii) decisions related to internal resource assignment.

## Pilot-API

The aim of the Pilot-API [2] is to provide an abstract, unified interface to PJ frameworks that adhere to the P\* Model, e.g. BigJob [1], DIANE and Condor/Condor-G (see figure).



The Pilot-API has been used to support a wide range of infrastructures, applications types and usage modes. For example, we successfully utilized the Pilot-API and different distributed infrastructures with BFAST, a genome sequencing application. BFAST is very I/O sensitive, which leads to an I/O bottleneck if many BFAST CUs run on the same shared file system. The Pilot-API enables applications to scale to different infrastructures in such cases. The following figure shows how the Pilot-API is used to run BFAST interoperably on different production and research infrastructures.



## References:

- [1] A. Luckow, L. Lacinski, and S. Jha, "SAGA BigJob: An Extensible and Interoperable Pilot-Job Abstraction for Distributed Applications and Systems," in IEEE/ACM CCGrid, 2010
- [2] Pilot API, <https://github.com/saga-project/BigJob/blob/master/pilot/api/>, 2012.