



J. Caballero<sup>a</sup>, J. Hover<sup>a</sup>, P. Love<sup>b</sup>, G. A. Stewart<sup>c</sup> for the ATLAS Collaboration  
<sup>a</sup>Brookhaven National Laboratory, <sup>b</sup>University of Lancaster, <sup>c</sup>CERN

The ATLAS experiment at the CERN LHC is one of the largest users of grid computing infrastructure, which is a central part of the experiment's computing operations. Considerable efforts have been made to use grid technology in the most efficient and effective way, including the use of a pilot job based workload management framework. In this model the experiment submits 'pilot' jobs to sites without payload. When these jobs begin to run they contact a central service to pick-up a real payload to execute.

The first generation of pilot factories were usually specific to a single VO, and were bound to the particular architecture of that VO's distributed processing.

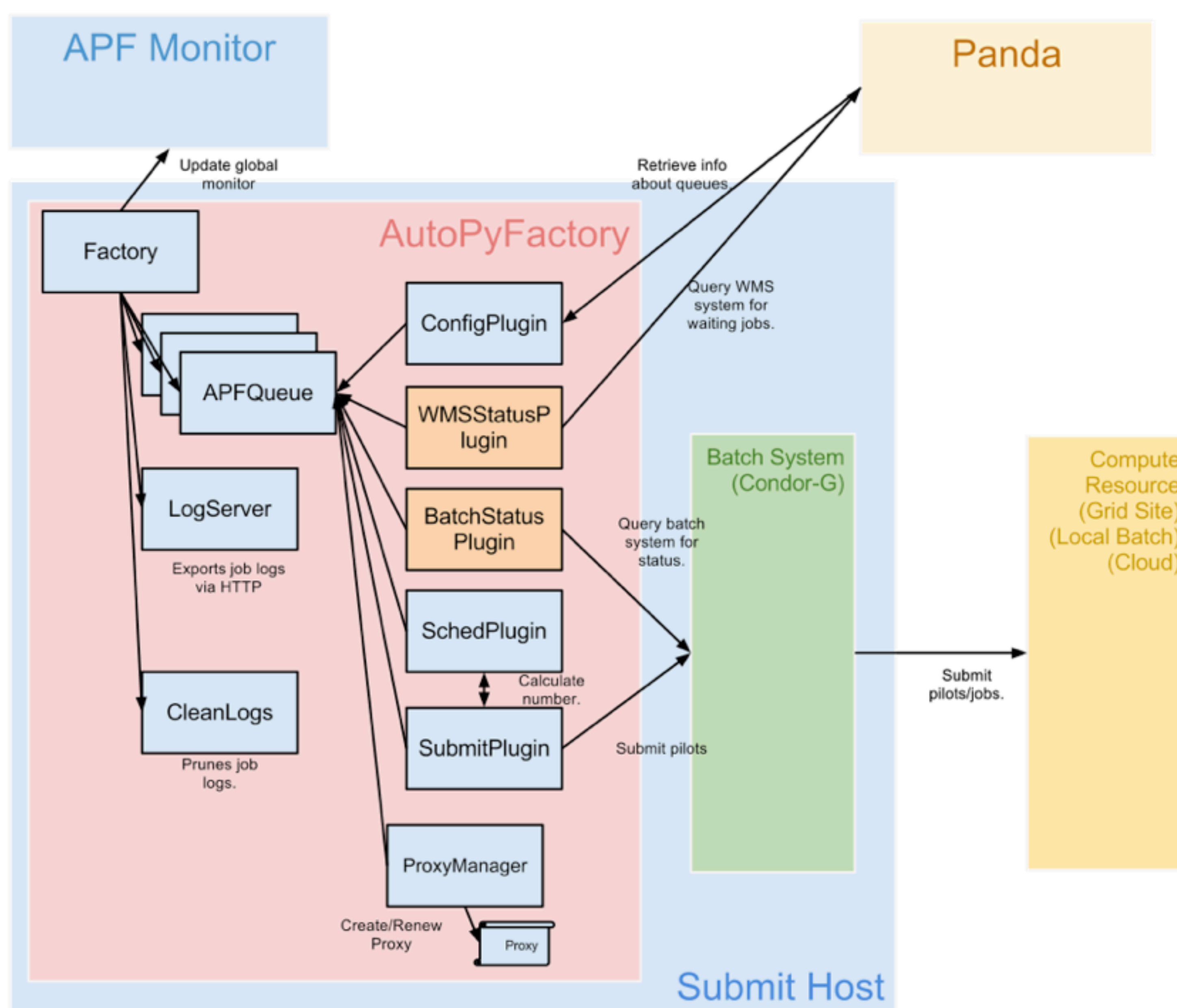
A second generation provides factories which are more flexible, not tied to any particular VO, and provide new and improved features such as monitoring, logging, profiling, etc. In this paper we describe this key part of the ATLAS pilot architecture, a second generation pilot factory, AutoPyFactory.

AutoPyFactory has a modular design and is highly configurable. It is able to send different types of pilots to sites and exploit different submission mechanisms and queue characteristics. It is tightly integrated with the PanDA job submission framework, coupling pilot flow to the amount of work the site has to run. It gathers information from many sources in order to correctly configure itself for a site and its decision logic can easily be updated.

Integrated into AutoPyFactory is a flexible system for delivering both generic and specific job wrappers which can perform many useful actions before starting to run end-user scientific applications, e.g., validation of the middleware, node profiling and diagnostics, and monitoring.

AutoPyFactory also has a robust monitoring system that has been invaluable in establishing a reliable pilot factory service for ATLAS.

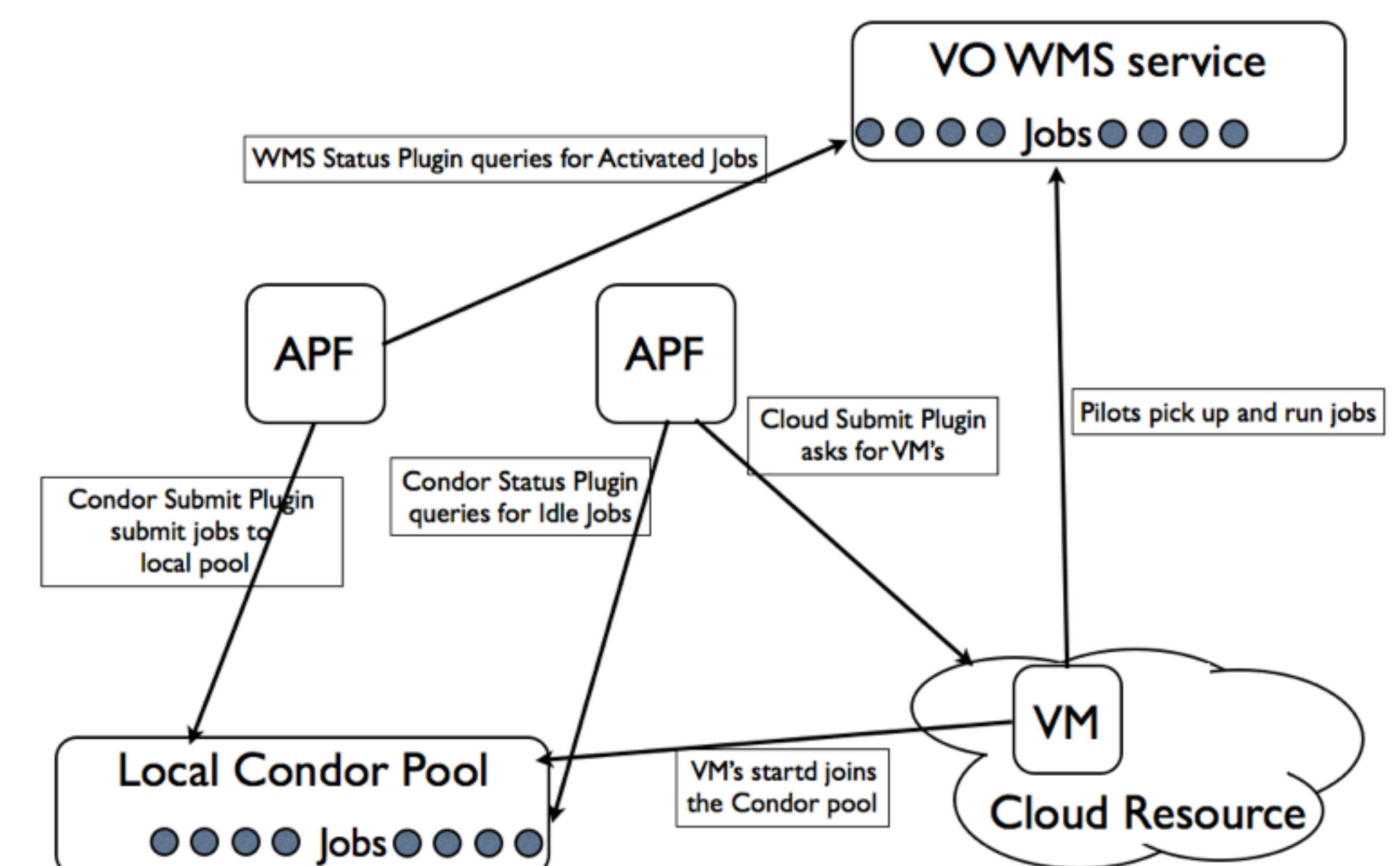
## Design



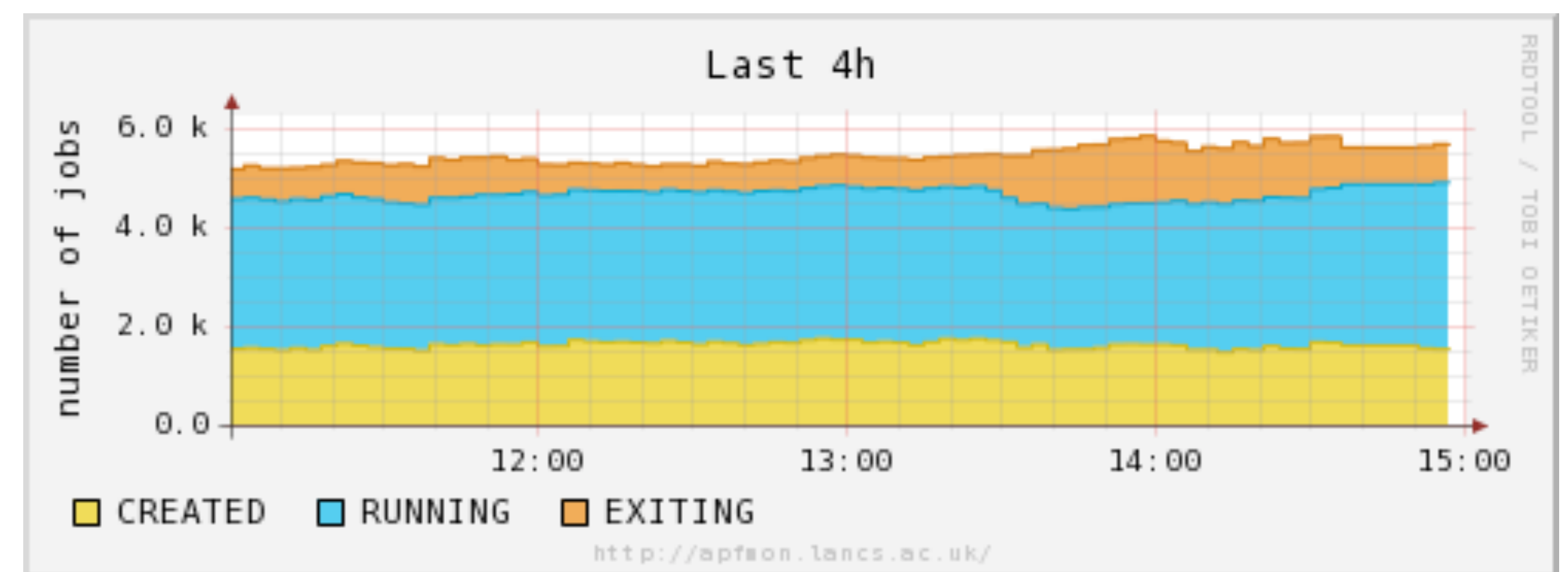
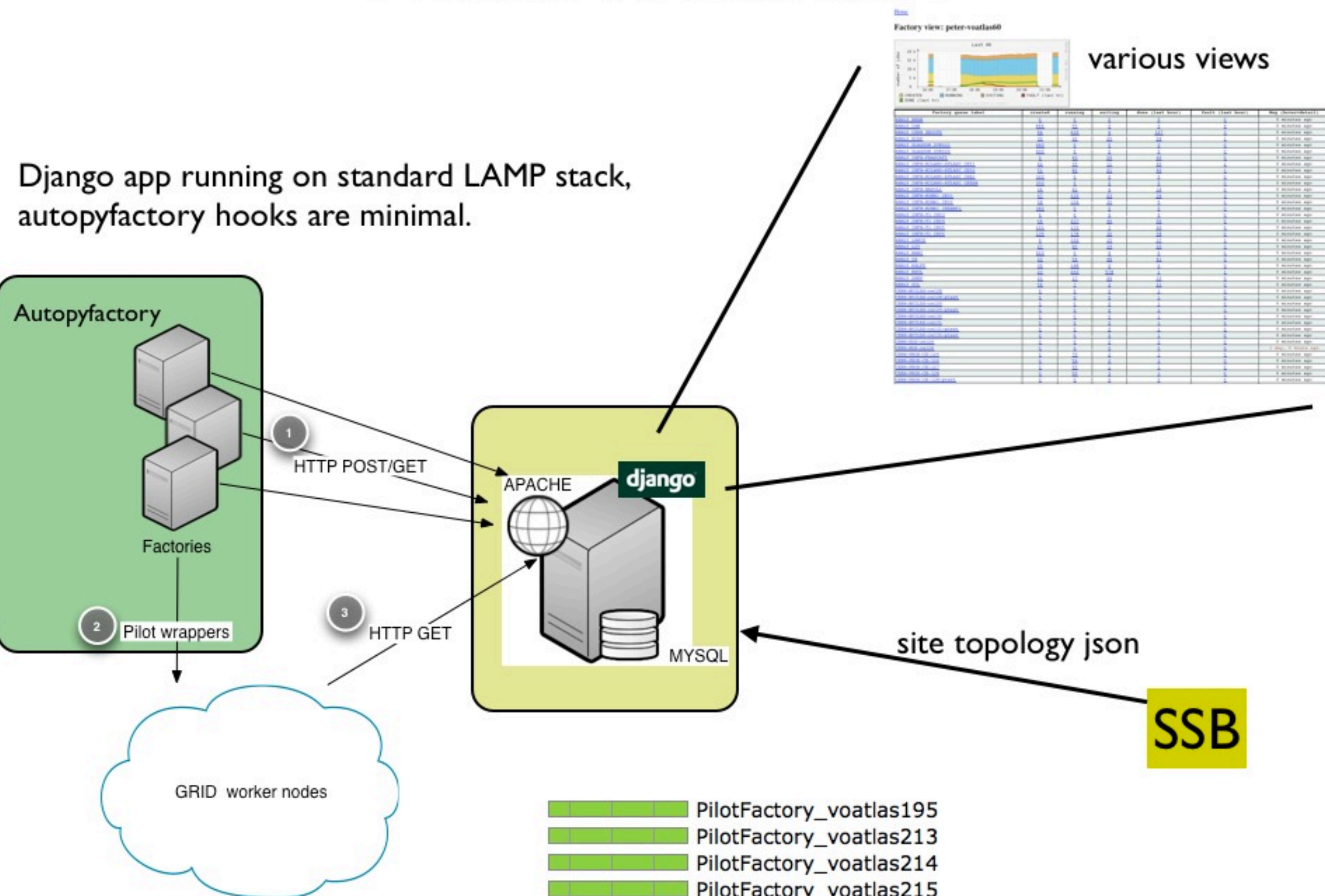
## Plug-ins Architecture

**WMS Status Plug-in** Queries VO WMS for its current state, e.g., how many jobs activated? Running? *E.g., PandaWMSStatusPlugin.*  
**Batch Status Plug-in** Queries local batch system (e.g., Condor-G or Condor) for submitted jobs state info (pending, failed, running, etc.) *E.g., CondorBatchStatusPlugin.*  
**Sched Plug-in** Implements the algorithm which decides exactly how many pilots to submit each cycle. *E.g., ActivatedPlugin, FixedPlugin, NQueuePlugin.*  
**Submit Plug-in** Creates the submit file and issues batch submit command(s). *E.g., CondorGT2BatchSubmitPlugin, CondorEC2BatchSubmitPlugin.*  
**Configuration Plug-in** Retrieves extra configuration object to be merged at run time to the local configuration. *E.g., PandaConfigPlugin.*

## Cloud Computing with AutoPyFactory



## Monitor Architecture



Factory queue label	created	running	exiting	done (1 hr)	fault (1 hr)	last msg
ANALY_BNL_ATLAS_1	10	41	1	0	0	7 minutes a
ANALY_TEST-APF2	0	4	5	0	0	7 minutes a
BNL_ATLAS_1	3	19	1	0	0	1 minute a
BNL_CVMFS_1	0	60	1	0	0	0 minutes a
BNL_ITB_Test1	0	0	0	0	0	1 week, 6 days a
BNL_TEST-APF	0	0	0	0	0	1 week, 5 days a
TEST2	348	0	0	0	0	1 minute a