Dynamic parallel ROOT facility clusters on the Alice Environment
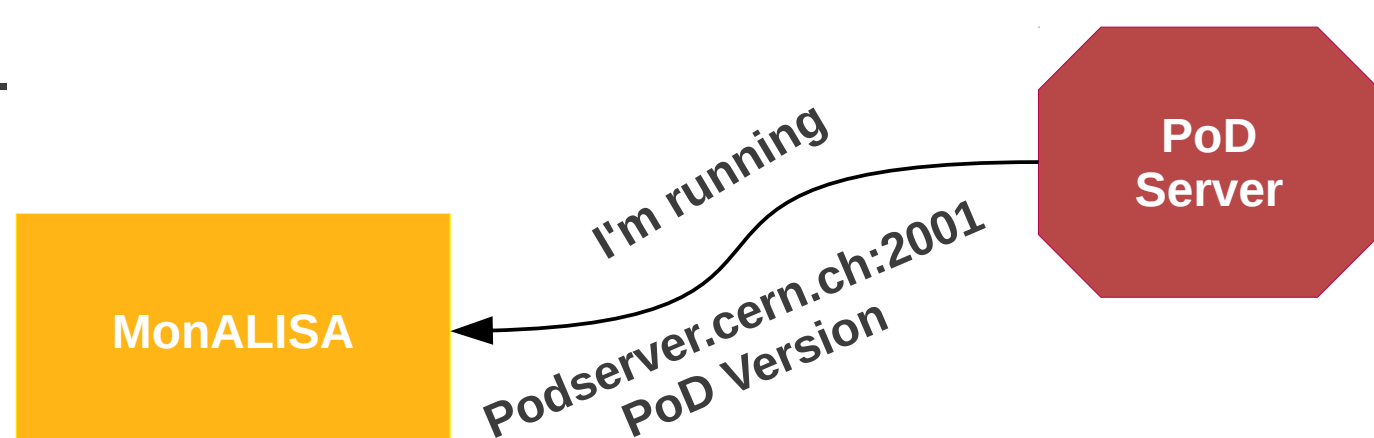
Cinzia Luzzi (University of Ferrara, CERN), Latchezar Betev (CERN), Federico Carminati (CERN), Costin Grigoras (CERN), Pablo Saiz (CERN), Anar Manafov (GSI)

The ALICE collaboration has developed a production environment (AliEn) that implements full set of the Grid tools enabling the full offline computational work-flow of the experiment, simulation, reconstruction and data analysis, in a distributed and heterogeneous computing environment. In addition to analysis on the Grid, ALICE uses a set of local interactive analysis facilities installed with the Parallel ROOT Facility (PROOF). PROOF enables physicists to analyze medium-sized (order of 200-300 TB) data sets on a short time scale. The default installation of PROOF is on a static dedicated cluster, typically 200-300 cores . This well-proven approach, has its limitations, more specifically for analysis of larger datasets or when the installation of a dedicated cluster is not possible. Using a new framework called PoD (Proof on Demand), PROOF can be used directly on Grid-enabled clusters, by dynamically assigning Interactive nodes on user request. **The integration of Proof on Demand in the AliEn framework provides private dynamic PROOF clusters as a Grid service. This functionality is transparent to the user who will submit interactive jobs to the AliEn.**

PoD Server within the AliEn System

The user requests a PoD Server by submitting a job on AliEn. A minimum and a maximum number of workers and a maximum of waiting time to get the maximum number of workers can be specified on the jdl. The detection of the request for a PoD server is a trigger for AliEn to start the server on a dedicated AliEn Computing Element. As soon as the PoD server is running it starts to advertise itself to AliEn by publishing information to MonALISA and it stops when it has acquired the predefined set of workers.
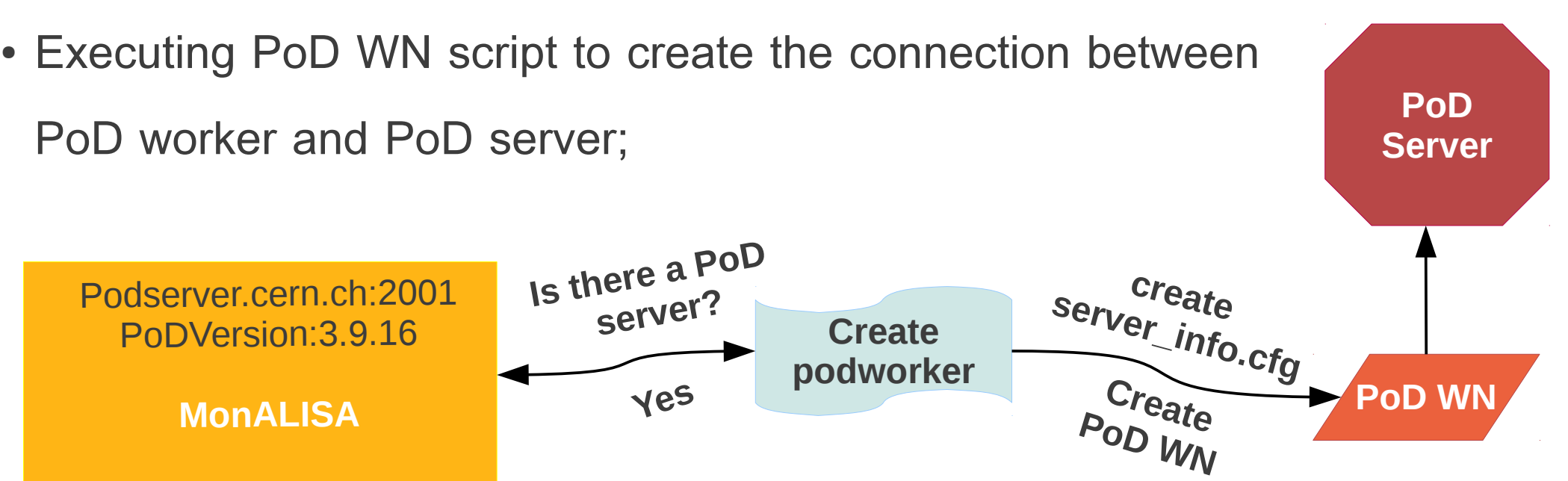

After a predefined amount of idle time, the server will shut down automatically.

PoD Workers within the AliEn System

A routine integrated in the AliEn system checks MonALISA to find a PoD server available. Once detected a request from the PoD server, the PoD worker is started on the standard worker node (WN) :

- Creating PoD WN package with the information about the PoD server (hostname, port number, PoD Version) taken from MonAlisa
- Executing PoD WN script to create the connection between PoD worker and PoD server;

If there is no PoD server available, the AliEn WN executes normal jobs. The JobAgent stops to check MonALISA as soon as it gets a free PoD server or as soon as the predefined timeout expires. In the last case it starts to execute normal jobs.
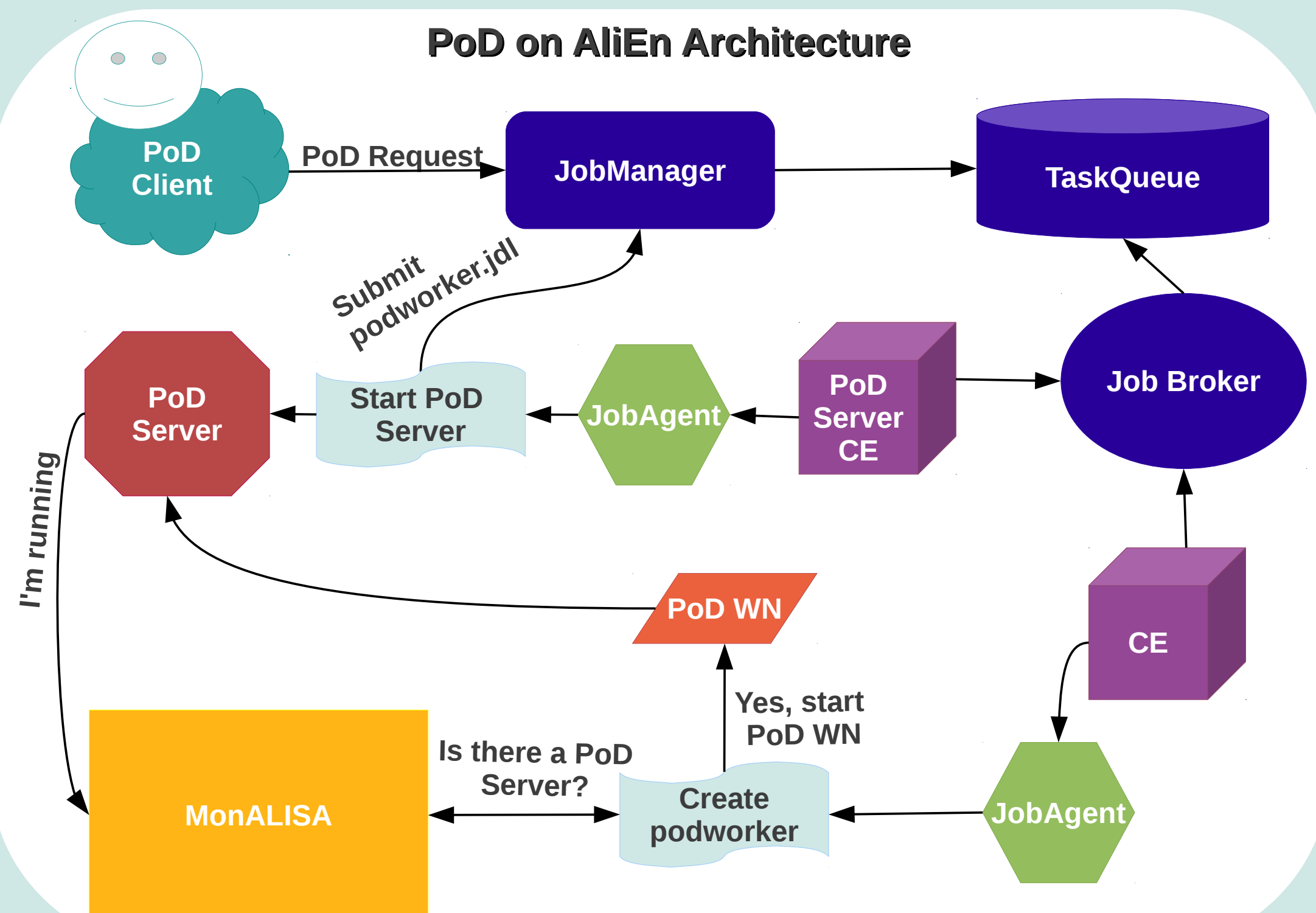
Easy Request PoD Server JDL Example

```
Executable = "StartPoDServer.sh";
Partition = "PoDServer";
Packages = {"ROOT","PoD"};
Arguments = 5; → Number of Workers
```

Easy to install, maintain and use

After the user's request, PoD will be installed directly from AliEn with predefined PoD packages. An important feature of PoD WN is that it starts its own xproofd daemon, in this way we can control and handle each PoD worker individually. The procedure to create and connect the PoD WNs with the PoD server will run in the VOBox. In this way no additional installation is required!

PoD on AliEn Architecture

Outlook

Alien-PoD worker acquires a list of free server nodes on demand. Currently, the first free PoDserver that is on the MonALISA list is the one that will execute the next PoD job. This part will be improved with an optimized procedure to choose the available PoD server, for example the nearest one can be chosen in order to reduce the network traffic and system overheads. In orden to select the closest PoD server, each MonALISA VOBox service could start a web server with a similar layout as alimonitor (PoDMonALISA). The PoD server will send its information to the closest PoDMonALISA service in the environment and the CreatePoDWorker procedure will ask for the information to the closest PoDMonALISA service too.